Crossbow

MoteView Users Manual

Revision B, June 2006

PN: 7430-0008-04

www.xbow.com

© 2003-2006 Crossbow Technology, Inc. All rights reserved.

Information in this document is subject to change without notice.

Crossbow, MoteWorks, MICA, TrueMesh and XMesh are registered trademarks of Crossbow Technology, Inc. Other product and trade names are trademarks or registered trademarks of their respective holders

Table of Contents

1 Introduction 1
1.1 Wireless Mesh Networking Overview 1
1.2 MoteView Overview 2
1.3 Supported Sensor Boards and Mote Platforms 2
1.4 Supported Mote Software Applications 3
2 Installation 7
2.1 Supported PC Platforms and Operating Systems 7
2.2 Additional Software Requirements 7
2.3 Installation Steps 8
3 Application Quick Start 11
3.1 Connecting to the Demo Database on Your PC 11
3.2 Connecting to an Active Sensor Network Using an MIB510 or MIB600 12
3.3 Connecting to a Database (Local or Remote) Server or a Stargate 14
3.4 Auto-discovery of Nodes 14
4 User Interface Functional Description 15
4.1 *MoteView* at a Glance 15
4.2 Node List 15
4.3 Visualization Tabs 15
5 Menus 25
5.1 File 25
5.2 Tools 26
5.3 Units 40
5.4 Windows 41
5.5 Help 41
5.6 Toolbar Menu 41
5.7 Dialogs 42
5.8 Server Messages 44
6 Server Administration 45
6.1 General configuration 45
6.2 XServe Logging 45
7 Database Administration 47
7.1 PostgreSQL 47
7.2 SQL 47

7.3 Database Tools 49
8 Over-the-Air-Programming (OTAP) using MoteConfig 50
8.1 OTAP Preparation using *MoteConfig* Utility 50
8.2 OTAP using *MoteConfig* Utility 52

About This Document

The following annotations have been used to provide additional information.

NOTE

Note provides additional information about the topic.

☑ EXAMPLE

Examples are given throughout the manual to help the reader understand the terminology.

IMPORTANT

This symbol defines items that have significant meaning to the user

WARNING

The user should pay particular attention to this symbol. It means there is a chance that physical harm could happen to either the person or the equipment.

The following paragraph heading formatting is used in this manual:

1 Heading 1

1.1 Heading 2

1.1.1 Heading 3

This document also uses different body text fonts (listed in Table 0-1) to help you distinguish between names of files, commands to be typed, and output coming from the computer.

Table 0-1. Font types used in this document.

Font Type	Usage
`Courier New Normal`	Sample code and screen output
`Courier New Bold`	Commands to be typed by the user
Times New Roman Italic	TinyOS files names, directory names
Franklin Medium Condensed	Text labels in GUIs

1 Introduction

1.1 Wireless Mesh Networking Overview

Wireless sensor networks have attracted a wide interest from industry due to their diversity of applications. Sensor networks are pervasive by nature; the number of nodes in a network is nearly boundless. Therefore, a key to realizing this potential is multi-hop mesh networking, which enables scalability and reliability.

A mesh network is really a generic name for a class of networked embedded systems that share several characteristics including:

- **Multi-Hop**—the capability of sending messages peer-to-peer to a base station, thereby enabling scalable range extension;
- **Self-Configuring**—capable of network formation without human intervention;
- **Self-Healing**—capable of adding and removing network nodes automatically without having to reset the network; and
- **Dynamic Routing**—capable of adaptively determining the route based on dynamic network conditions (e.g., link quality, hop-count, gradient, or other metric).

When combined with battery power management, these characteristics allow sensor networks to be long-lived, easily deployed, and resilient to the unpredictable wireless channel. With mesh networking, the vision of pervasive and fine-grained sensing becomes reality.

A wireless network deployment is composed of the three distinct software tiers:

- The **Client Tier** provides the user visualization software and graphical interface for managing the network. Crossbow provides free client software called *MoteView* that bundles software from all three tiers to provide an end-to-end solutions.
- The **Server Tier** is an always-on facility that handles translation and buffering of data coming from the wireless network and provides the bridge between the wireless motes and the internet clients. *XServe* and *XOtap* are server tier applications that can run on a PC or Stargate.
- The **Mote Tier**, where *XMesh* resides, is the software the runs on the cloud of sensor nodes forming a mesh network. The *XMesh* software provides the networking algorithms required to form a reliable communication backbone that connects all the nodes within the mesh cloud to the server.

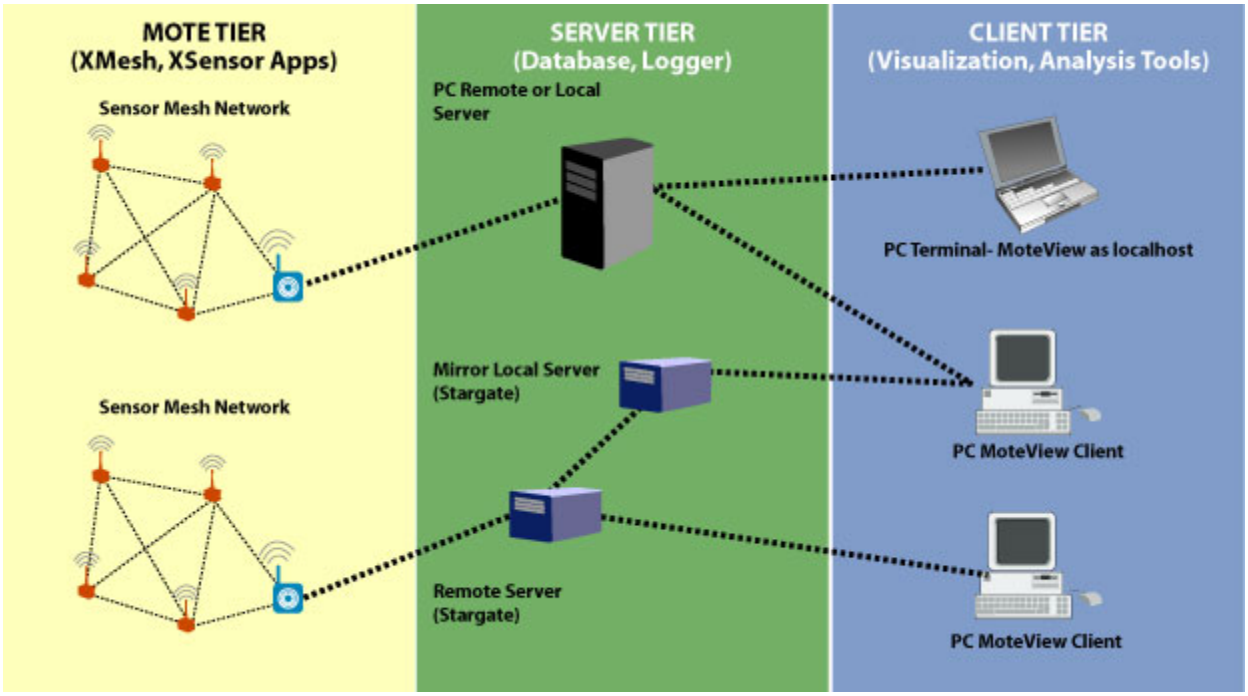


Figure 1-1. Software framework for a wireless sensor network

1.2 MoteView Overview

MoteView is designed to be an interface ("client tier") between a user and a deployed network of wireless sensors. *MoteView* provides users the tools to simplify deployment and monitoring. It also makes it easy to connect to a database, to analyze, and to graph sensor readings.

Figure 1-1 depicts a three-part framework for deploying a sensor network system. The first part is the Mote layer or sensor mesh network. The Motes are programmed with *XMesh*/TinyOS firmware ("application") to do a specific task: e.g., microclimate monitoring, asset tracking, intrusion detection, etc. The second layer or Server tier provides data logging and database services. At this layer sensor readings arrive at the base station (e.g., MIB510, MIB520, MIB600, or Stargate) and are stored on a server or Stargate. Finally, the third part is the client tier in which software tools provide visualization, monitoring, and analysis tools to display and interpret sensor data. The purpose of this document is to explain the features of *MoteView* and to provide information on the supported Mote layer applications, Mote platforms, and sensor boards.

1.3 Supported Sensor Boards and Mote Platforms

All of Crossbow's sensor and data acquisition boards are also supported by *MoteView* (see Table 1-1). *MoteView* supports the MICA-series platforms of wireless sensor network hardware, including the MICA2, MICA2DOT, and MICAz Motes (see Table 1-2). In addition, sensor integrated platforms such as the security/intrusion detection system based on the MSP Motes and the environmental monitoring system (based on the MEP Motes) can be deployed and monitored (see Table 1-3).

Table 1-1. Sensor (MTS series) and data acquisition boards supported by MoteView and their plug-and-play compatible Mote platforms

Sensor and Data Acquisition Boards	Mote Platforms		
	MICAz	MICA2	MICA2DOT
MTS101	✓	✓	
MTS300/310	✓	✓	
MTS410	✓		
MTS400/MTS420	✓	✓	
MTS450	✓	✓	
MTS510			✓
MDA100	✓	✓	
XBW-DA100	✓		
MDA300	✓	✓	
MDA320	✓	✓	
XBW-DA325	✓		
MDA500			✓
MTP400		✓	

Table 1-2. Mote processor/radio (MPR) platforms supported by MoteView

Mote Platforms	Model Number(s)	RF Frequency Band(s)
MICAz	MPR2420	2400 MHz to 2483.5 MHz
	MPR2600	2400 MHz to 2483.5 MHz
MICA2	MPR400	868 MHz to 870 MHz; 903 MHz to 928 MHz
	MPR410	433.05 to 434.8 MHz
	MPR420	315 MHz (for Japan only)
	MPR600	868 MHz to 870 MHz; 903 MHz to 928 MHz
MICA2DOT	MPR510	868 MHz to 870 MHz; 903 MHz to 928 MHz
	MPR510	433.05 to 434.8 MHz
	MPR520	315 MHz (for Japan only)

Table 1-3. Sensor integrated (MEP, MSP) platforms supported by MoteView

Sensor Integrated Mote Platforms	Description of Usage
MEP410	Microclimate and ambient light monitoring
MEP510	Temperature and humidity monitoring
MSP410	Physical security and intrusion detection

1.4 Supported Mote Software Applications

MoteView can support a number of different firmware applications running on the Motes that make up the mesh network:

NOTE: All the Mote applications mentioned below can be programmed through the *MoteConfig* tool described in Section 5.2.6, if the application is precompiled. Only the *XMesh*-enabled applications come pre-compiled with *MoteView*. Each Mote must be programmed with a unique identification number ("node ID") when deploying a network of greater than one node.

XMesh is Crossbow's multihop mesh networking protocol that has various options including low-power listening, time synchronization, sleep modes, any-to-base and base-to-any routing. All of our sensor and data acquisition boards are supported with *XMesh* enabled applications. The tables below are a summary of the *XMesh* applications the corresponding sensor boards.

Table 1-4. Pre-compiled MICAz XMesh applications available in the MoteView

MICAz Mote (MPR2400 and MPR2600)	
Board Model	**Binary file name**
MTS boards	
MTS101	*XMTS101_2400_<mode>.exe*
MTS300CA	*XMTS300CA_2400_<mode>.exe*
MTS300CB	*XMTS300CB_2400_<mode>.exe*
MTS310CA	*XMTS310CA_2400_<mode>.exe*
MTS310CB	*XMTS310CB_2400_<mode>.exe*
MTS400	*XMTS420_2400_<mode>.exe*
MTS410	*XMTS410_2400_<mode>.exe*
MTS420	*XMTS420_2420_hp.exe*
MTS450	*XMTS450_2400_<mode>.exe*
MDA board	
MDA100CA	*XMDA100CA_2400_<mode>.exe*
MDA100CB	*XMDA100CB_2400_<mode>.exe*
XBW-DA100CA	*XBW-DA100CA_2400_hp.exe*
XBW-DA100CB	*XBW-DA100CB_2400_hp.exe*
MDA300	*XMDA300_2400_<mode>.exe*
MDA320	*XMDA320_2400_<mode>.exe*
XBW-DA325	*XBW-DA325_2400_hp.exe*
Base Station (common to all boards)	
XMeshBase_2400_<mode>.exe	

Table 1-5. Pre-compiled MICA2 XMesh Applications Available in the MoteView

MICA2 Mote (MPR4x0, x = 0, 1, or 2 and MPR600)	
Board Model	**Binary file name**
MTS boards	
MTS101	*XMTS101_xxx_<mode>.exe*
MTS300CA	*XMTS300CA_xxx_<mode>.exe*
MTS300CB	*XMTS300CB_xxx_<mode>.exe*
MTS310CA	*XMTS310CA_xxx_<mode>.exe*
MTS310CB	*XMTS310CB_xxx_<mode>.exe*
MTS400	*XMTS400_xxx_<mode>.exe*
MTS420	*XMTS420_xxx_<mode>.exe*
MTS410	*XMTS410_xxx_<mode>.exe*
MTS450	*XMTS450_xxx_<mode>.exe*
MDA board	
MDA100CA	*XMDA100CA_xxx_<mode>.exe*
MDA100CB	*XMDA100CB_xxx_<mode>.exe*
MDA300	*XMDA300_xxx_<mode>.exe*
MDA320	*XMDA300_xxx_<mode>.exe*
Base Station (common to all boards)	
XMeshBase_xxx_<mode>.exe	

xxx = 315, 433, or 915. <mode> = hp or lp. hp = high power mesh networking. lp = low-power mesh networking via low-power listening and time synchronized data transmissions.

Table 1-6. Pre-compiled MICA2DOT XMesh Applications Available in MoteView

MICA2DOT Mote (MPR5x0, x = 0, 1, or 2)	
Board Model	**Binary file name**
MTS boards	
MTS510	*XMTS510_xxx_<mode>.exe*
MDA boards	
MDA500	*XMDA500_xxx_<mode>.exe*
Base Station (common to all boards)	
XMeshBase_Dot_xxx_<mode>.exe	

xxx = 315, 433, or 915. <mode> = hp or lp. hp = high power mesh networking. lp = low-power mesh networking via low-power listening and time synchronized data transmissions.

Table 1-7. Pre-compiled MSP XMesh applications available in MoteView

MSP410 Mote Security Package	
MSP410_433_base.exe	To be used on an MPR410 MICA2 Mote connected to the base station
MSP410_433_hp.exe	For the MSP410 Mote

Table 1-8. Pre-compiled MEP XMesh applications available in MoteView

MEP410 Mote Environmental Package	
MEP410_433_<mode>.exe	For the MEP410 and an MPR410 MICA2 Mote when assigned a node ID of 0 ("base node").
MEP510 Mote Environmental Package	
MEP510_433_<mode>.exe	For the MEP510 and an MPR510 MICA2DOT Mote when assigned a node ID of 0 ("base node").

<mode> = hp or lp. hp = high power mesh networking. lp = low power mesh networking via low-power listening and time synchronized data transmissions.

2 Installation

2.1 Supported PC Platforms and Operating Systems

MoteView is supported on the following platforms:

- *Windows XP Home*
- *Window XP Professional*
- *Windows 2000 with SP4*

The screen resolution must be at least 800 × 600 or the interface would require scrollbars.

2.1.1 PC Interface Port Requirements

There are three different interface port requirements depending on the gateway platform used as the base station in a sensor network.

1. For an **MIB510** serial gateway: an RS-232 serial port.
2. For an **MIB520** USB gateway: a USB port.
3. For an **MIB600** Ethernet gateway: A wired Ethernet or 802.11 wireless card only if the MIB600 is on a LAN with wireless access.
4. For a **Stargate** gateway or other server: A wired Ethernet, an 802.11 wireless card only if the Stargate has a wireless modem or is on a LAN with wireless access, or a cellular modem for wireless Internet access.

2.2 Additional Software Requirements

For the application to run the following additional components are required:

- PostgreSQL 8.0 *database service*
- PostgreSQL *ODBC driver*
- *Microsoft .NET framework*

The installation files for these components are included on the *MoteView* installation CD.

2.2.1 PostgreSQL

All the visualization tools in *MoteView* require being connected to a database. This database can reside on your PC ("localhost"), a remote server, or a Stargate. The size of this database is bound by the storage available on the system.

The installation of PostgreSQL will automatically install and configure a local *PostgreSQL* 8.0 database on the machine when that option is checked. The installation requires administrative privileges on the system, including the ability to create a new user called *postgres* (for *PostgreSQL*).

IMPORTANT: If your PC is to host both the server and client layer functions, then running the *PostgreSQL* database service is required to use *MoteView*. However, if you are accessing a server or Stargate that is running *XServe/PostgreSQL*, then you don't need to run the service on your PC.

2.3 Installation Steps

Shut down all the programs running on your computer.

1. Insert the *MoteView Support* CD into the computer's CD drive.
2. Double-click on ***MoteViewSetup.exe*** from *MoteView* folder.
3. At the **Welcome to the MoteView Setup Wizard** window, click **Next>**.
4. Select the desired installation directory, then click **Next>** (see Figure 2-1).

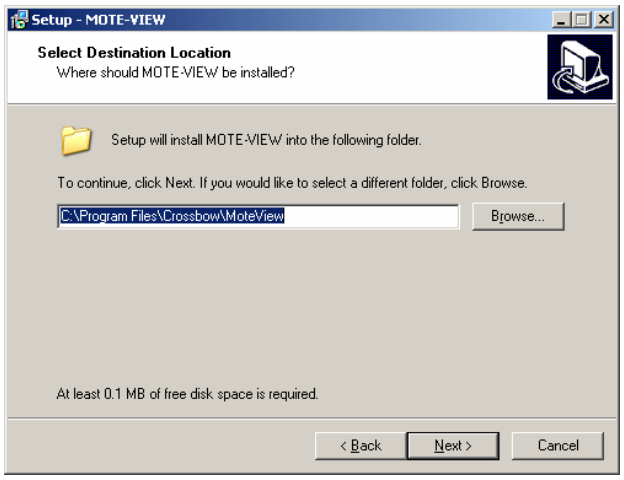

Setup - MOTE-VIEW
Select Destination Location
Where should MOTE-VIEW be installed?
Setup will install MOTE-VIEW into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Program Files\Crossbow\MoteView
Browse...
At least 0.1 MB of free disk space is required.
< Back
Next >
Cancel

Figure 2-1. Step 4: Destination folder select and confirm screen

5. Select the desired Start Menu folder name and click **Next>**.

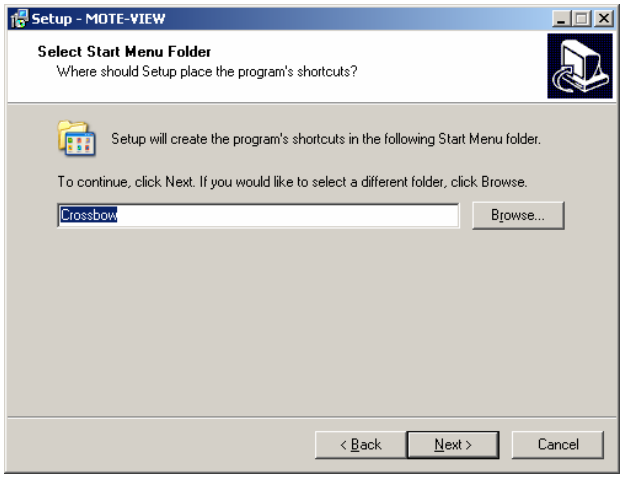

Setup - MOTE-VIEW
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
Crossbow
Browse...
< Back
Next >
Cancel

Figure 2-2. Step 5: Screenshot for selecting the Start Menu folder for MoteView

6. Select all available installation tasks and click **Next>** (see Figure 2-3).

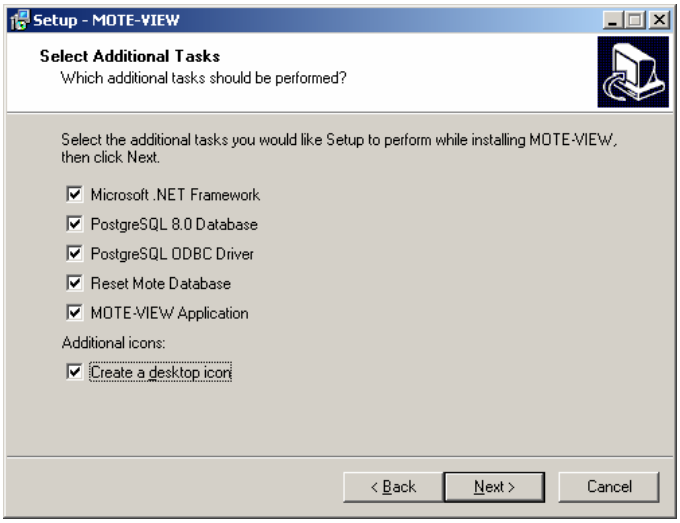

Setup - MOTE-VIEW
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing MOTE-VIEW, then click Next.
Microsoft .NET Framework
PostgreSQL 8.0 Database
PostgreSQL ODBC Driver
Reset Mote Database
MOTE-VIEW Application
Additional icons:
Create a desktop icon
< Back
Next >
Cancel

Figure 2-3. Step 6: Select all options if installing MoteView for the first time

7. Confirm your selections and click **Install.**

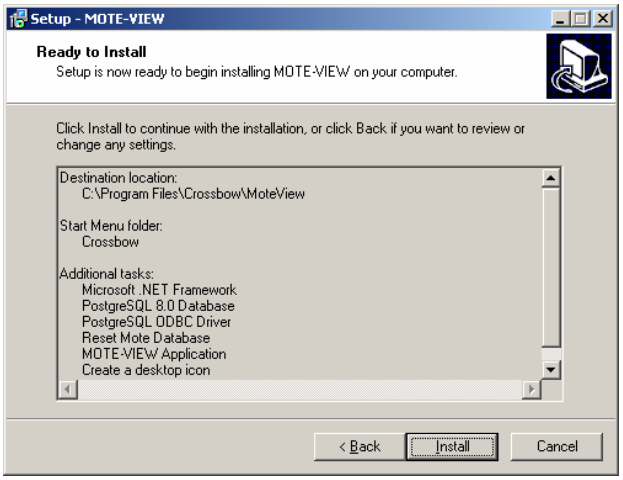


Figure 2-4. Step 7: Confirm you selections with this screen

Execute the .NET framework and *PostgreSQL* OBDC installation prompts if applicable and follow the instructions.

: NOTE: If you have a version of MoteView prior to v1.0 you may get a **PostgreSQL 8.0.0-rc1** window (Figure 2-5) when installing the database. Record the exact error message in *C:\Program Files\PostgreSQL\8.0.0-rc1\tmp\initdb.log*. In most cases you can simply click the **OK** button and proceed. However, in other cases you may have to do the following:

a. Uninstall *PostgreSQL 8.0* from **Start>Control Panel>Add or Remove Programs**

b. Remove the *C:\Program Files\PostgreSQL* directory manually through the Windows Explorer utility (**Start>**Right Click**>Explore**).

c. Resintall *MoteView* with all checked **except** the .NET Framework.

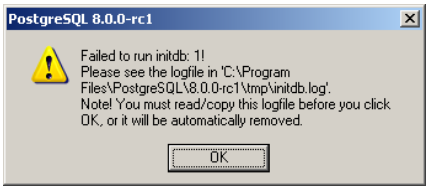

PostgreSQL 8.0.0-rc1
Failed to run initdb: 1!
Please see the logfile in 'C:\Program Files\PostgreSQL\8.0-rc1\tmp\initdb.log'.
Note! You must read/copy this logfile before you click OK, or it will be automatically removed.
OK

Figure 2-5. Error window that may occur during the PostgreSQL Installation

8. Please note that if you receive an MDAC warning such as the one pictured below, you may ignore this warning and continue on with the installation. The warning is simply notification that your operating system contains an MDAC version that is newer than what *MoteView* is installing.

Setup - MoteView
Installing
MDAC 2.8 RTM
MDAC 2.8 RTM
MDAC 2.8 RTM is incompatible with this version of Windows. All of its features are currently part of Windows.
To exit, click Cancel. For more information, click Details.
Don't display this message again
Continue
Cancel
Details
Finish
Cancel

Figure 2-6. MDAC 2.8 RTM incompatibility message that may appear during .NET installation. This warning can be ignored by clicking on Cancel

9. When the Setup Wizard has finished it will ask if you want to start *MoteView*. You may start *MoteView* now, but in some cases it may ask you to restart your computer first.

10. If you have restarted or at any time after you can start *MoteView* by double- clicking on the *MoteView* icon or by going to the installation folder (the default install folder is ***C:\Program Files\Crossbow\MoteView***) and clicking on ***MoteView.exe***.

3 Application Quick Start

Once a sensor network is running and *MoteView* is installed on a PC, there is minimal configuration necessary to start collecting data from the sensor network.

IMPORTANT: There are two ways to repair a corrupt database installation:

1. Run the installer again and select only "Reset Mote Database"
2. Open a File Manager, navigate to C:\Program Files\Crossbow\MoteView, and execute "resetdb.cmd".

3.1 Connecting to the Demo Database on Your PC

During the installation of *MoteView* a static database was included to make it possible to demonstrate *MoteView's* features without having to be connected to an active sensor network or to be connected to a remote server/database. The steps described here also apply to viewing data collected during a connection to an active sensor network.

1. Click on the green connect icon , in the upper left hand corner or click on **File > Connect to Remote Database/Stargate.** The **Database Server Configuration** will appear (see Figure 3-1).

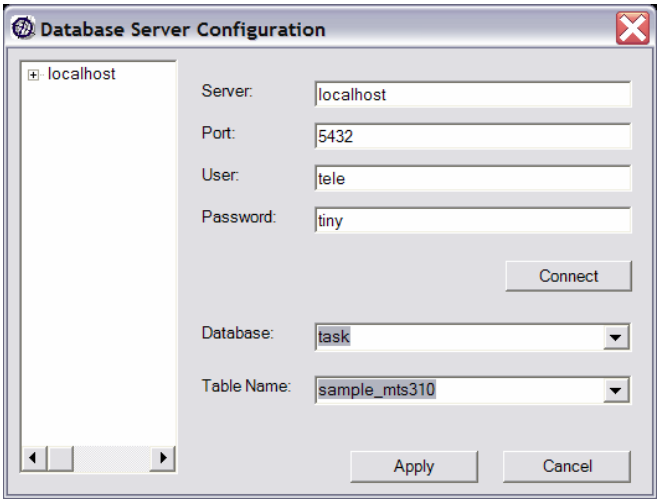


Figure 3-1. Database Server Configuration window which provides access to remote or local servers and databases

2. Check that the pull down item next to **Database = task** and that **Table Name = sample_mts310.**
3. Click on Apply. *MoteView* will then access the table in the database task. Afterwards the screen will display the data from the database table file `sample_mts310`.

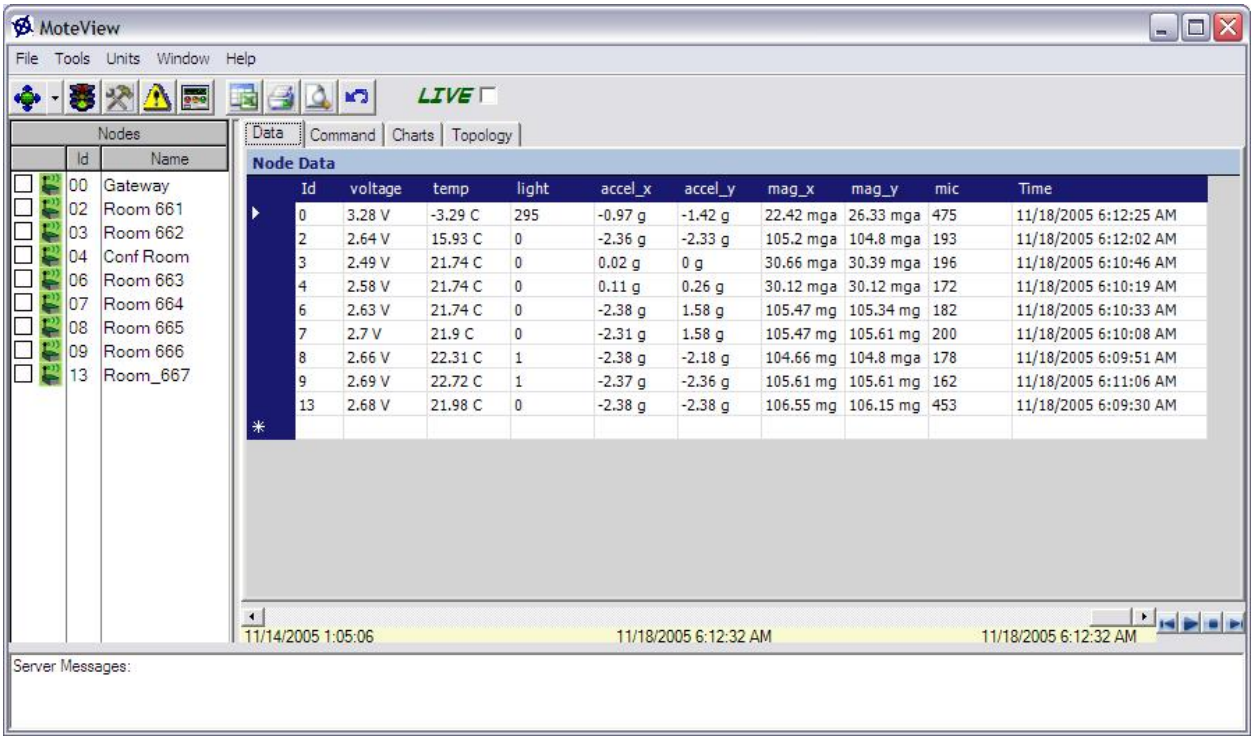


Figure 3-2. After MoteView connects to a database or after starting a data log, data starts to appear in the MoteView *Data* view

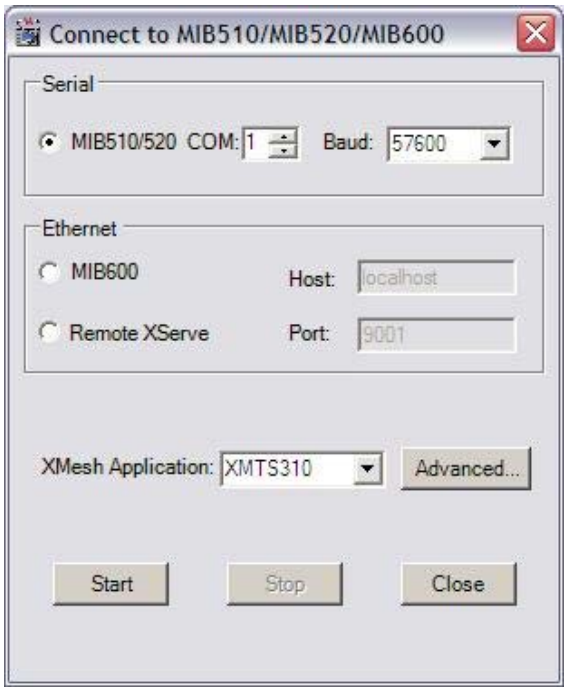
3.2 Connecting to an Active Sensor Network Using an MIB510 or MIB600

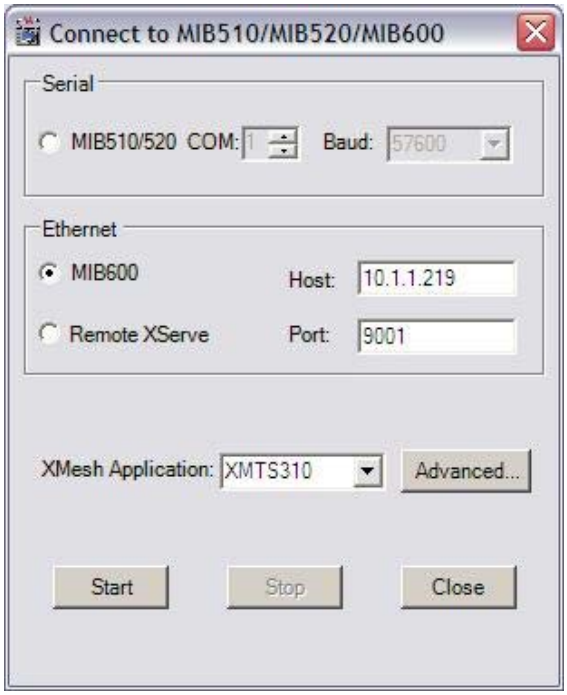
The first step typically is to connect to either a gateway device—MIB510, MIB600, or Stargate—which will send data to your PC ("localhost server") or a server or Stargate that's running a Postgres database.

1. On the menu bar click on **File > Connect to MIB510/520/600**.
2. If using a MIB510/MIB520, make sure that the **Serial Port: COM** is set to the correct port number and that baud is **57600**.
3. If using a MIB600, select the button next to **Host** and in the box next to it enter the IP address of the MIB600. The **Port** should default to **mib600**.

☑ EXAMPLE 3-1:

Screenshots of the **Connect to MIB510/MIB520/MIB600** popup window are shown below. On the left is an example of connecting to an MIB510 on COM1. The baud rate is set to **mica2** which is equivalent to a data rate of 57600 bps. On the right is an example of connecting to an MIB600 assigned with IP address 10.1.1.219, which is connected to the LAN.





4. Choose the *XMesh* application that matches to that which the Mote has been programmed.

If you have an *XMesh* Application…	Then choose one of the following pull down choices
Default. Use when you don't see the *XMesh* application	(none)
XMTS101_<freq>_<mode>	XMTS101
XMTS300_<freq>_<mode>	XMTS300
XMTS310_<freq>_<mode>	XMTS310
XMTS400_<freq>_<mode>	XMTS400
XMTS410_<freq>_<mode>	XMTS410
XMTS420_<freq>_<mode>	XMTS420
XMTS450_<freq>_<mode>	XMTS450
XMTS510_<freq>_<mode>	XMTS510
XMDA100_<freq>_<mode>	XMDA100
XBW-DA100_<freq>_<mode>	XBW-DA100
XMDA300_<freq>_<mode>	XMDA300
XMDA320_<freq>_<mode>	XMDA320
XBW-DA325_<freq>_<mode>	XBW-DA325

XMDA500_<freq>_<mode>	XMDA500
XMEP410_<freq>_<mode>	XMEP-SYS
XMEP510_<freq>_<mode>	XMEP-SYS
XMSP410_<freq>_hp.exe	XMSP410

IMPORTANT: The **Log to Database** option must be selected when connecting to a MIB510, MIB520 or MIB600 for viewing live sensor network data.

5. If you are not able to receive data, you may also need to select the **Live** check box on the main *MoteView* screen if it has not been previously selected. Also check to see if data is coming in from your nodes by viewing the **Server Messages** pane at the bottom of your *MoteView* display or by looking at the *XServe* shell.

3.3 Connecting to a Database (Local or Remote) Server or a Stargate

To connect to the Sensor Network,

1. Click on the green **Connect** icon , in the toolbar menu. As necessary fill out the information in top four boxes the **Database Server Configuration** dialog window. The minimal configuration requires information for the **Server**, **Port**, **User**, and **Password**.

NOTE: Users should be advised that the **Server**, **Port**, **Username**, and **Password** fields are preset to reasonable defaults and should **not** have to be changed.

2. Click the **Connect** button to connect *MoteView* to the given server.
3. Select the **Database** and then **Table Name** from the drop down menu.
4. Click on the **Apply** button to store your selections and begin viewing data from the database. This configuration will be stored in a short pull down list to the right of the **Connect** icon. This quick connect list will remember the last table used for each **Server-Database** pair.

3.4 Auto-discovery of Nodes

New nodes will also appear in the upper left hand corner of the **Topology** view. These nodes can be dragged to their correct position on the topology map and those placements can be saved to the configuration file in the database.

4 User Interface Functional Description

4.1 *MoteView* at a Glance

MoteView has four main user interface sections:

1. Toolbar / Menus: Allows the user to specify actions and initiate command dialogs.
2. Node List: Shows all known nodes in a deployment and health status summary.
3. Visualization Tabs: Enables the user to view the sensor data in various ways.
4. Server Messages: Displays a log of server events and incoming messages.

4.2 Node List

The Node List displays all the known nodes in a deployment.

Table 4-1. Features and icon properties of the node list

Node List Feature	Description
Checkbox next to node id	Select nodes to be plotted in the Charts view
Change node properties	Double click on a node in the list or right click and choose **Properties**.
Sort data	Click on the column header. Static display only when not in "live" mode.
Add nodes	Right click on any node or use the auto-detect feature as results arrive.
Gray Mote icon	No results received
Green Mote icon	Fresh results within the last 20 minutes
Moss (light green) Mote icon	Results stale by >20 minutes
Yellow Mote icon	Results stale by >40 minutes
Orange Mote icon	Results stale by >60 minutes
Red Mote icon	Results stale by more than a day

The time intervals for the node color changes can be edited from the dialog box that appears when you right-click on the node and select Set TimeRange.

4.3 Visualization Tabs

The visualization tabs provide three ways to view your sensor data. The main display of the user interface consists of the four tabs: 1) **Data**, 2) **Command**, 3) **Charts**, and 4) **Topology**.

4.3.1 Data View

The **Data** tab displays the latest sensor readings received for each node in the network (see Figure 4-1 for an example from the demo database). Any column of data can be sorted by left clicking the top. This allows you to sort by node ID, parent, temperature, voltage, last result time, or any other sensor reading.

The columns include: Id, <sensed value 1>, <sensed value 2>, ..., <sensed value *N*>, and Time. The sensor data is automatically converted into standard engineering units.

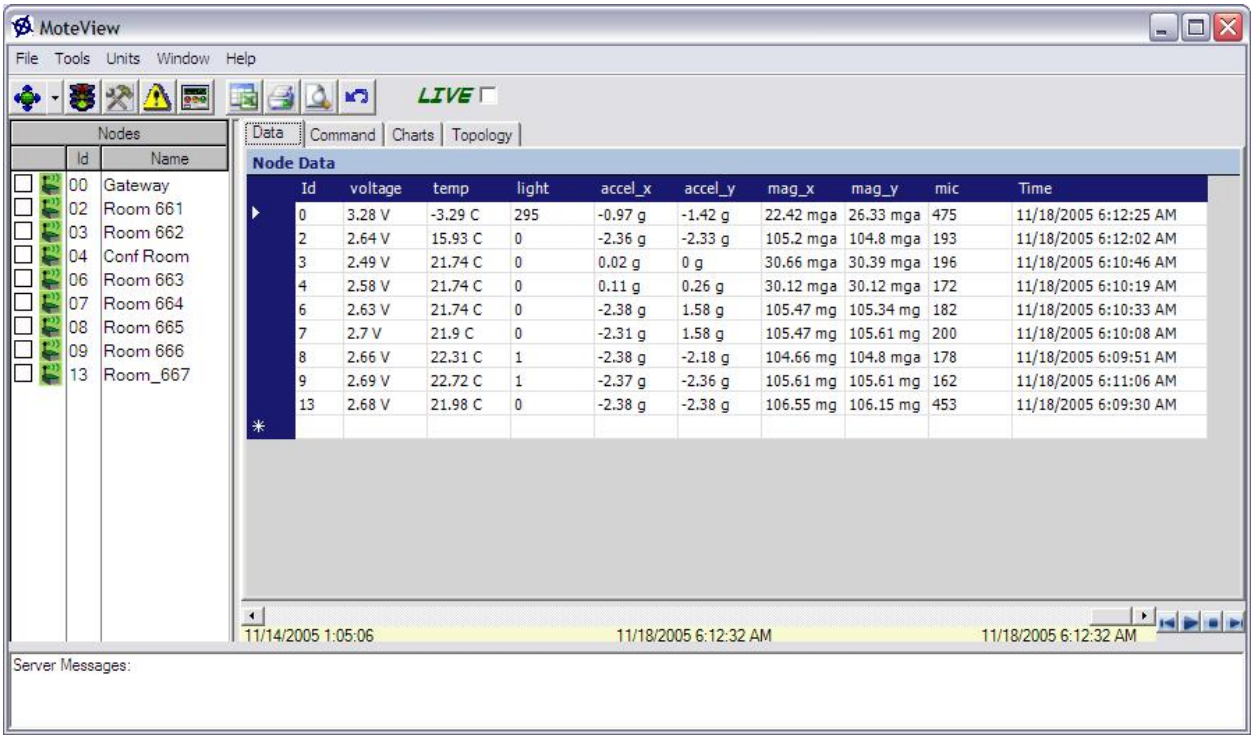


Figure 4-1. Screenshot of a demo database displayed in the *Data* view

4.3.2 Command

The **Command** tab provides the user with an ability to change different node parameters wirelessly.

NOTE: These commands do not get stored in the EEPROM and hence are volatile.

To change the System configuration do the following:

1. Click on **System** tab on the left side of the Command Window.
2. Select the Node ID and Group ID of your network. If changing the system parameters for all nodes, check the "**All Nodes**" checkbox.
3. From New Configuration, select either

   Data Rate – Sets the Update rate between 300 msec and 2147483647 msec

   Node ID – Changes the node ID between 1 and 65534

   Radio Power – Sets the RF power between 0 (min) and 255 (max)
4. Click on Set to send the specified commands to node (s) in your network.

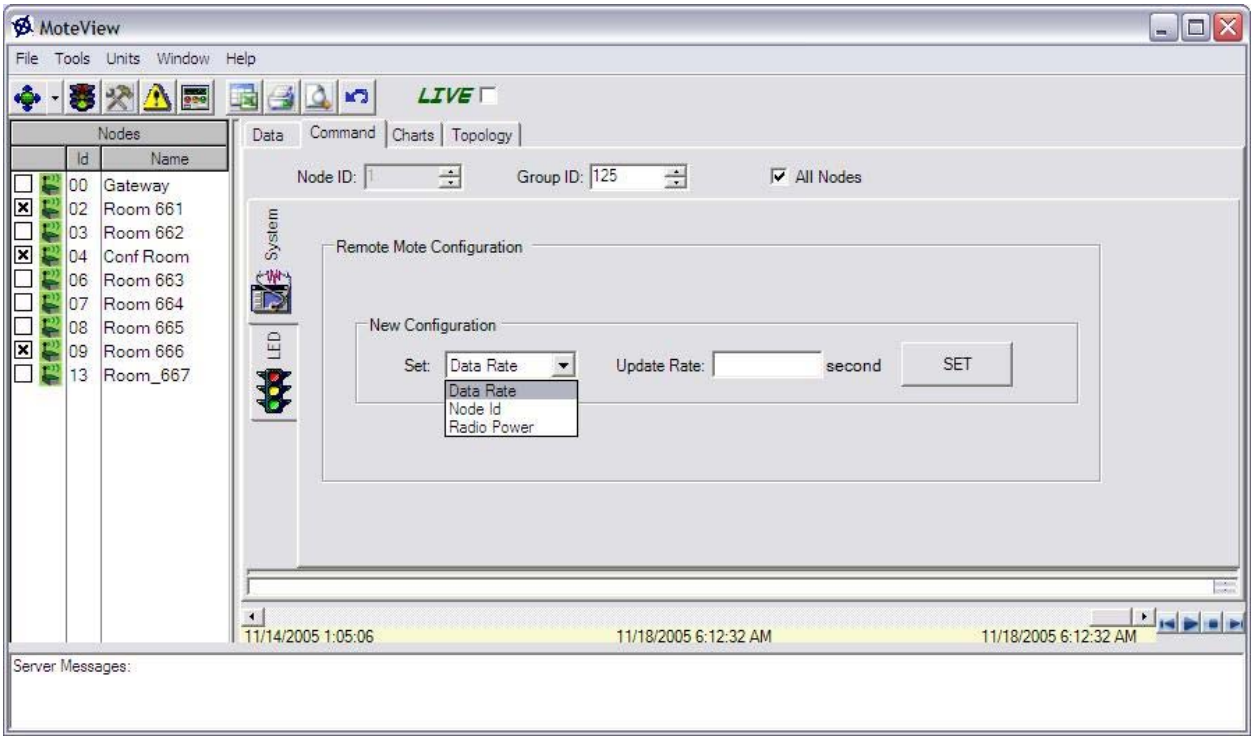


Figure 4-2. Screenshot of the System Configuration in *Command Tab*.

To change the LED status, do the following.

1. Click on **System** tab on the left side of the Command Window.
2. Select the Node ID and Group ID of your network. If changing the system parameters for all nodes, check the "All Nodes" checkbox.
3. Select the color of LED(s) you are interested (Red, Yellow or Green) and check

   ON – To turns the LED On.

   OFF – To turns the LED Off

   TOGGLE – To toggle the state of the LED
4. Click on Set to send the specified commands to node (s) in your network.
5. You can click SET ALL to activate all LEDs simultaneously.

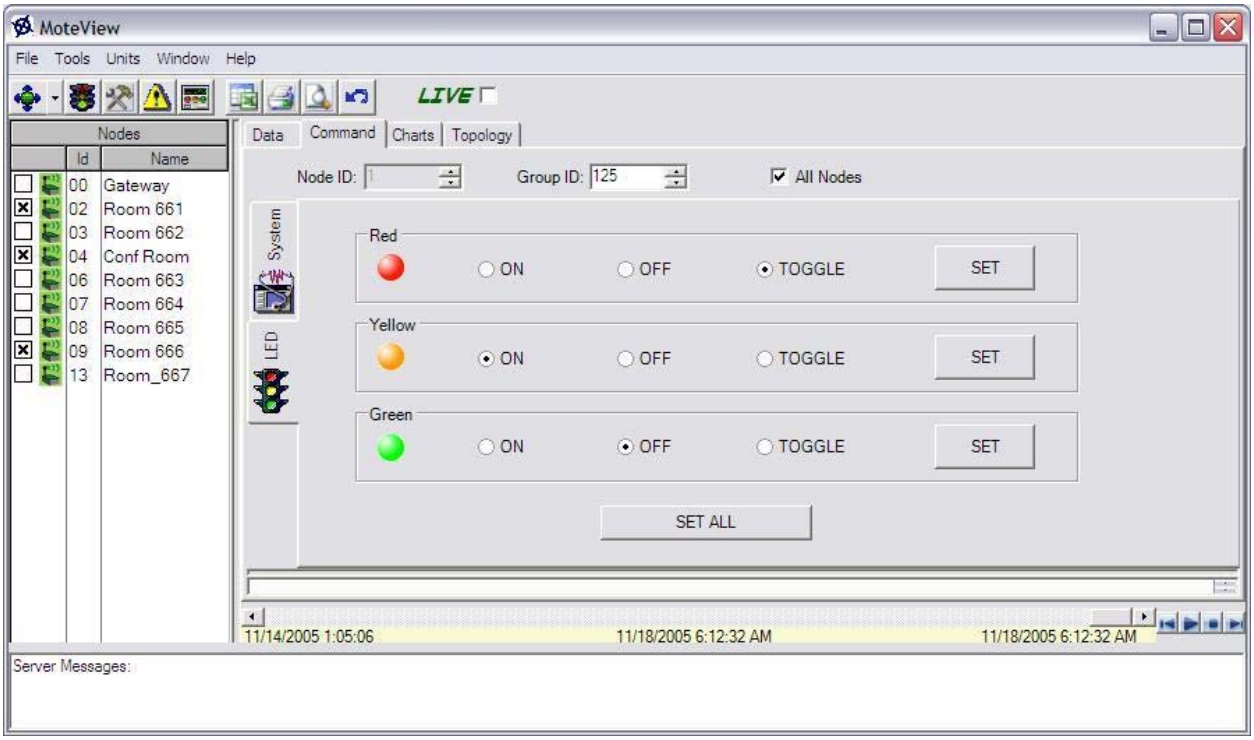


Figure 4-3. Screenshot of the LED actuation in *Command Tab*

NOTE: The Command feature is currently supported only for high power (hp) mesh networking applications.

4.3.3 Chart View

The **Chart** view provides the ability to generate graphs of a sensor reading against time for some set of nodes. The following features and constraints apply to the graphs plotted in this view.

- Up to 3 sensor types can be selected for plotting, i.e., 3 different graphs.
- Up to 24 different Nodes can be selected for plotting. A different color code is to be used for each node, with a legend on the right side of the window.
- The *y*-axis on the graph shows the time.
- The *x*-axis on the graph shows data in engineering units for the sensor readings.

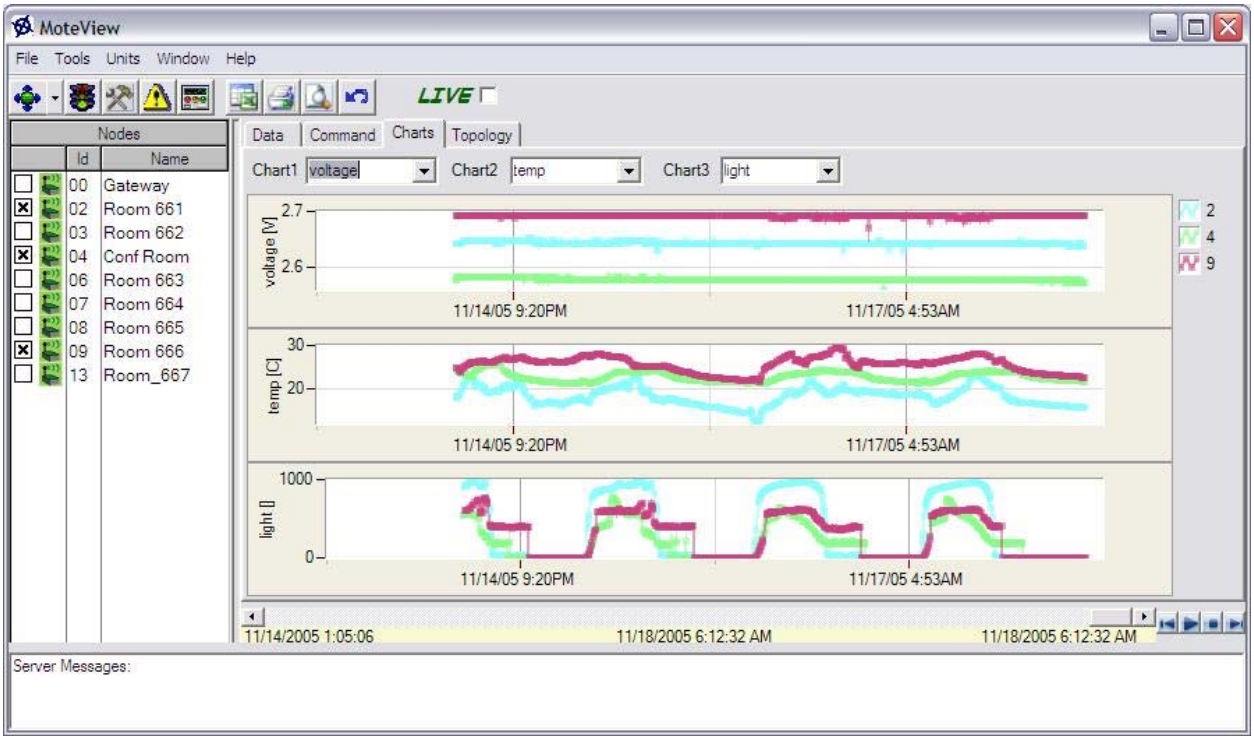


Figure 4-4. Screenshot of the demo database in *Chart* view

- The user can zoom into and pan through data, as instructed in Table 4-2, within each of the graphs independently of each other.

Table 4-2. How to zoom, pan, and reset in the Chart View.

Desired Action	Instructions
To zoom in	Hold down the "Shift" key. Left click and drag a region to zoom into. Release the mouse to complete the region selection. Release the "Shift" key.
To pan through data	Hold the "Ctrl" key Left click and drag a point within the chart to the new location. Release the mouse button.
To zoom out fully	Go to the Tools menu and select "Refresh View."
To undo a zoom	Hold down the "Shift" key Right click once for each level of undo
To undo a pan	Hold down the "Ctrl" key Right click once for each level of undo

4.3.4 Topology View

The **Topology** view shows a map of the network of Motes, including placement and parenting information. This allows the user to define and view a topology of their Mote deployment.

- New nodes will show up in the upper left hand corner.

- The user can drag nodes and place them at a new location on the map with a left click of the mouse. Node locations are stored in the database and are shared by all users of that database.

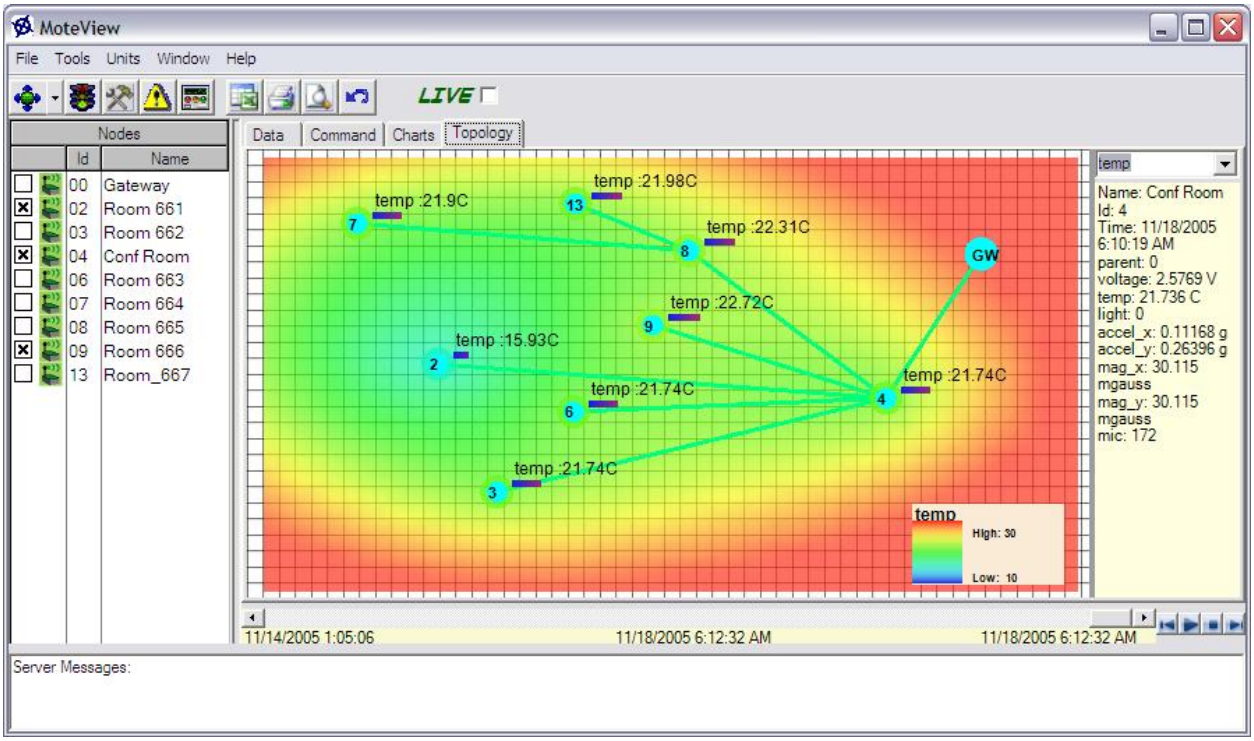


Figure 4-5. Screenshot of the demo database in Topology view

- A visualization pop-up menu will appear when the user right clicks the background bitmap.

Table 4-3. Items from the visualization pop-up menu that appears when right-clicking in the map area of the Topology view

Visualization Menu	Description
Add Node	Creates a new node at the current mouse location. A Mote properties dialog will appear to allow selection of node name and id number.
Arrange Nodes > Grid	Will automatically arrange all the nodes into a grid layout. This is useful when there are a large number of unplaced nodes.
Save Node Properties	Saves the current node placement and naming into the database.
Revert Node Properties	Reloads node placement and naming from the database
Visualization Properties	Allows the user to visualize temperature gradients or other Mote properties using a specific color pattern.
MSP Properties	Sets the orientation of quadrant 1 relative to image in the topology view and the draw radius of a PIR event
Load Bitmap	Allows users to select a bitmap from their file system to be the background of the map. Supported background image formats include: .bmp, .gif, .ico, and .jpg. The image will be automatically scaled to fit the available screen space.
Use Default Bitmap	Use the standard grid as the background of the topology view.

The Visualization Properties lets users customize the following:

Sensor Color Gradient – Allows users to specify the minimum and maximum sensor values and associate specific color with it.

1. Right-click on the Topology bitmap and select Visualization Properties.
2. Click on the Sensor tab of the Visualization Properties
3. Select a Sensor type from the drop-down list
4. Specify desired Maximum and Minimum sensor values
5. Click on the square colored box next to Maximum Color or Minimum Color
6. This will bring up another dialog box
7. Move the square marker on to choose the desired color and click OK.

The Sensor info also lets users display the sensor readings next to the node on the topology map and specify the font for the text.

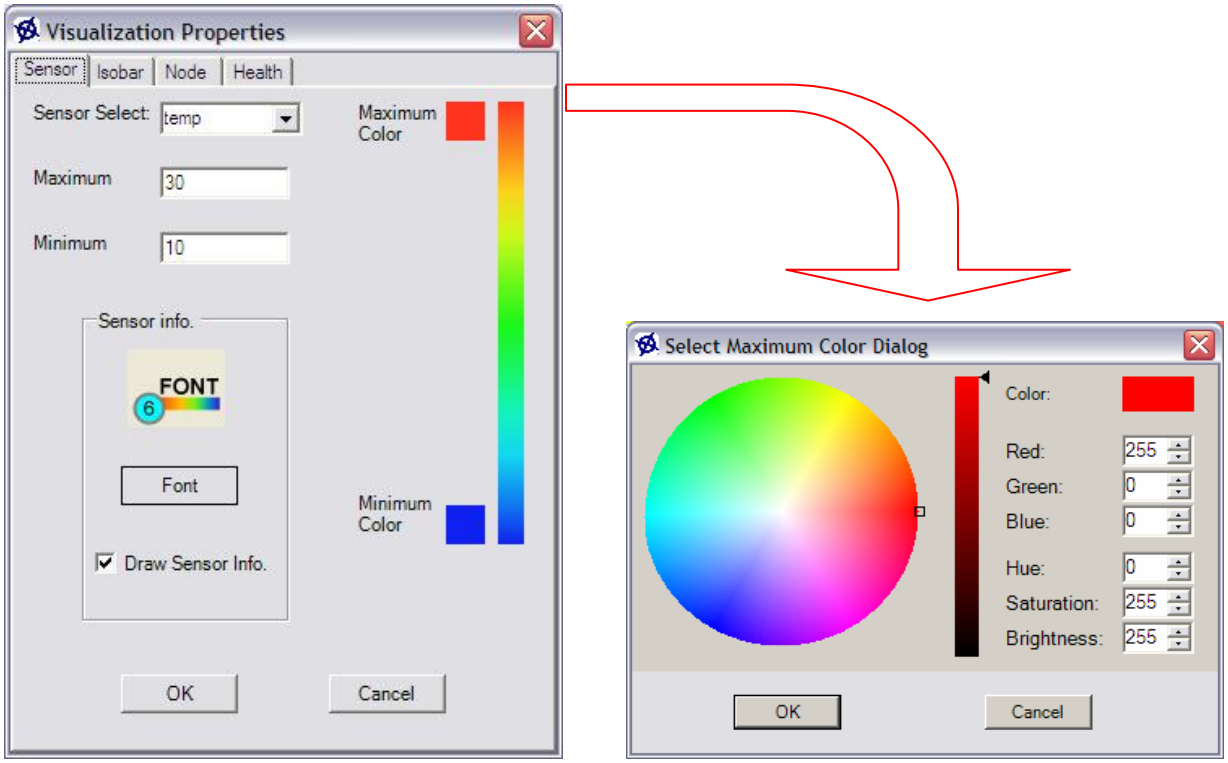


Figure 4-6. Screenshot for Sensor Gradient Visualization Dialog

Mote Visualization – Users can choose from 3 different display styles for nodes.

None – No nodes are displayed

BlackDot – A black dot is displayed to represent the node

MoteGlow – A colored circle is displayed along with the node ID.

Users can also check options for Draw Links between nodes and specifying whether Gateway has a sensor on it for gradient visualization.

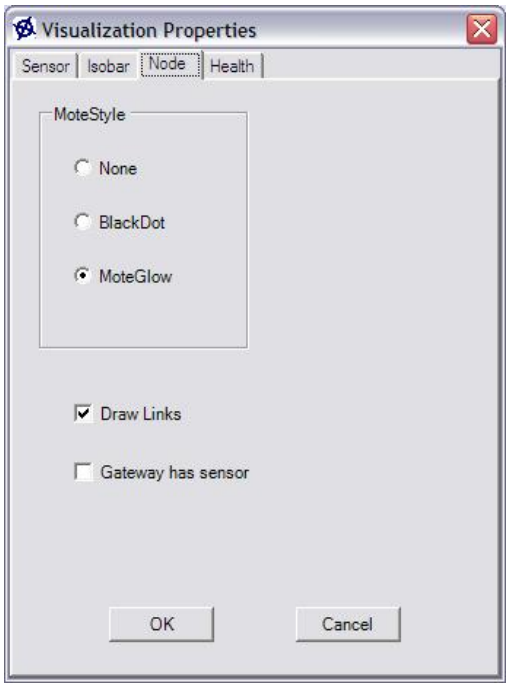


Figure 4-7. Screenshot for Mote Visualization Dialog

Isobar Visualization – Users can specify Isobar style

None – No Isobar visualization

Draw Gradient – Draws the color gradient based on the sensor readings;

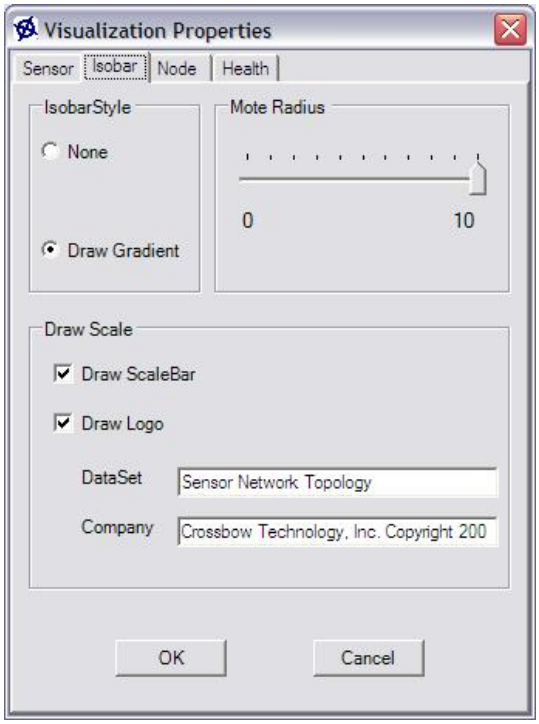


Figure 4-8. Screenshot for Isobar Visualization Dialog

Users can also specify the Mote Radius (specified as percentage of bitmap image size; 0=0%, 10=100%) for spread of gradient around the node; check options for displaying the ScaleBar Legend and drawing the Logo.

Health Visualization - Users can specify the time duration after which the link between the nodes goes grey. If a packet is not received from any after the specified minutes, the link would turn grey.

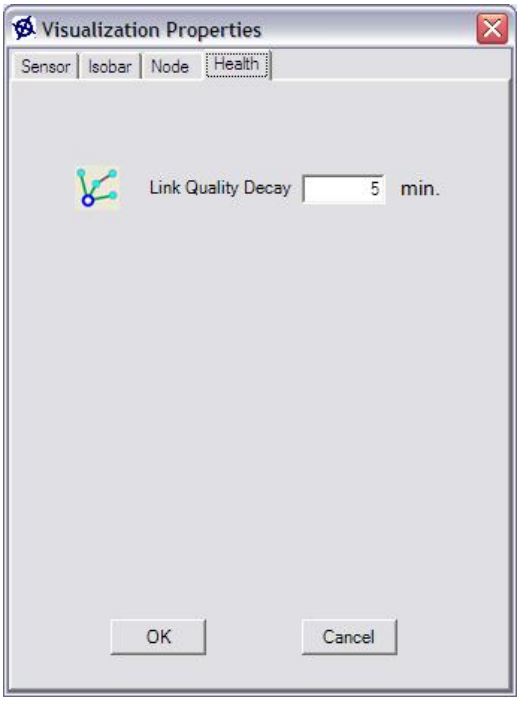


Figure 4-9. Screenshot for Link Quality Visualization Dialog

MoteView displays a colored circle around each node to indicate temperature values. The default values are listed in Table 4-4.

Table 4-4. *Color and Temperature Range Mapping*

Color	Temperature Range
Red:	> 30 °C
Orange:	26 ° to 30 °C
Yellow:	22 °C to 26 °C
Green:	18 °C to 22 °C
Blue:	< 18 °C

The user can bring up a menu of actions and access the Mote properties dialog with a right click of the mouse. By rolling over a node with the mouse, the user can bring up a status box with recent sensor readings of the Mote.

4.3.5 Live Mode

In order to have *MoteView* display data from an active sensor network, the users must check on the "**Live**" checkbox. In Live mode, *MoteView* refreshes the node list, charts, and topology views at a regular interval defined in the **Tools>Settings...** dialog. The default interval is 10 seconds.

: NOTE: The optimal setting for **Update Interval** is one half the sample rate used by the nodes.

5 Menus

There are two sections in the *MoteView* menu area. The top part is the horizontal arrangement of the five menu items: 1) **File**, 2) **Tools**, 3) **Units**, 4) **Window**, and 5) **Help**. Below this is a horizontally arranged icon toolbar, which provides quick access to some of the more common features of the *MoteView* application. From left to right, the toolbar provides buttons for database connection, data logging with *XServe*, *MoteConfig*, Alerts Manager, *MoteView* settings, export to CSV, print, and print preview, refreshing the view, and displaying live (vs. historical) sensor network data. This chapter will describe the features in each of these menu items.

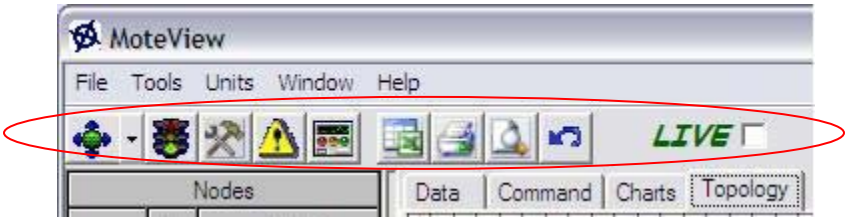


Figure 5-1. Screenshot showing a portion of the MoteView window to highlight the icon toolbar menu

5.1 File

5.1.1 Connect

This will bring up the Server Configuration dialog.

5.1.2 Export

You can export data in two different formats: XML or CSV (comma delimited text)

5.1.3 Print Preview

The Print Preview screen will display the expected output for the Print command. The current visualization tab is the one that is printed: Data, Chart, or Topology view. The magnifying glass icon allows you to zoom in on the preview. The printer icon will immediately print out the page to the default printer.

5.1.4 Print

The Print dialog is a standard Windows dialog that allows you to select the printer to which print output will be sent. The current visualization tab is the one that is being printed: Data, Chart, or Topology.

5.1.5 Exit

This menu option will exit the application and close *MoteView*. *MoteView* always attempts to save the current configuration when closed and restore that configuration the next time the application is started.

5.2 Tools

5.2.1 Settings

The Settings menu will bring up a dialog that allows the user to set MoteView global preferences.

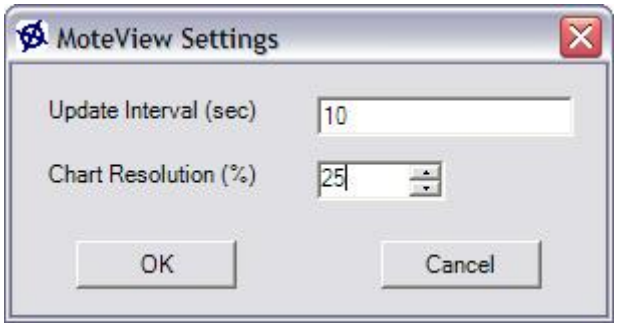


Figure 5-2. The Settings dialog to change the *Update Interval (sec)* and the *Chart Resolution (%)*

Currently this dialog provides access to two settings in *MoteView*:

Update Interval (sec): Specifies the length of time in seconds between database updates in "Live" mode.

NOTE: The optimal Update Interval is one half of the rate which the nodes are transmitting sensor data. This does **not** affect the rate at which the Motes are sending data into the network.

Chart Resolution (%): Specifies what percentage of data to draw from the database when creating charts. The lower the percentage, the faster the refresh and graphical update rates. That is, the trade off is between data resolution and increased graphics performance.

5.2.2 Add Node

This will bring up the Mote Properties dialog for creation of a new node. The user can select a unique node ID and name the node at this time.

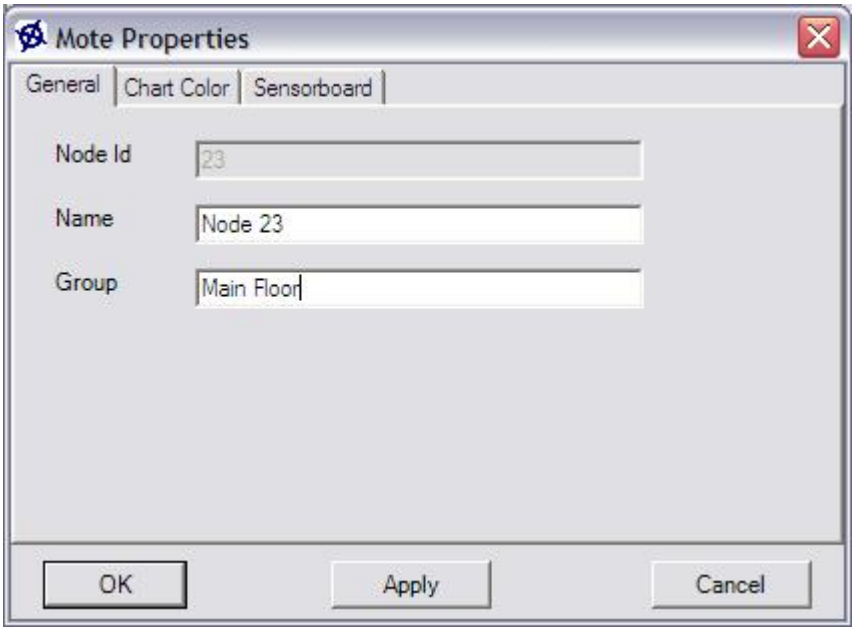


Figure 5-3. Screenshot of the Mote Properties Dialog Window

5.2.3 Refresh View

This will force a visual refresh of all visualization windows in *MoteView*. After clicking on this menu the **Nodes list**, the **Chart**, and **Topology** views will all update to display the latest information from the sensor network.

5.2.4 Log Data

The data **must** be actively logging into a database before live data can be displayed in MoteView. The **File > Connect to MIB510/MIB600** dialog (see Figure 5-4) provides a means to start and stop logging of Mote data on the user's PC ("**localhost**") using the *XServe* software tool.

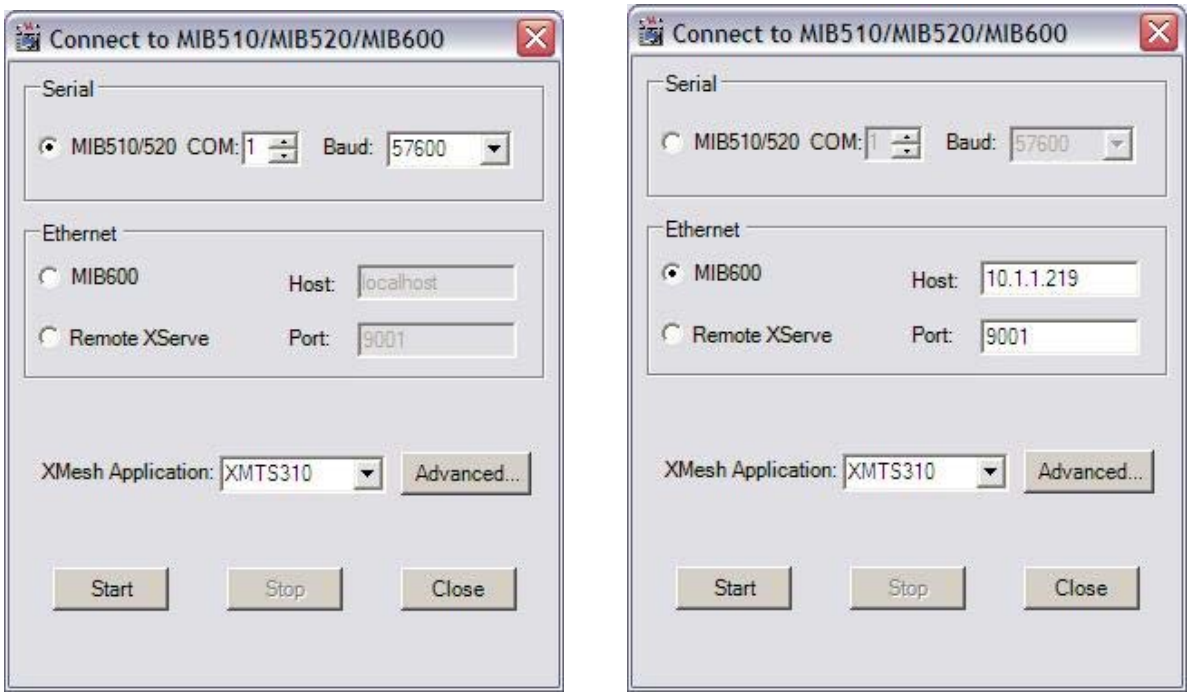


Figure 5-4. The Connect to MIB510/MIB520/MIB600 dialog.

5.2.5 Advanced Logging

By selecting the **Log to Database** check box and clicking on the **OK** button you will be able to receive live data from your nodes. Please note that you may also need to select the **Live** checkbox on the main *MoteView* screen if it has not been previously selected.

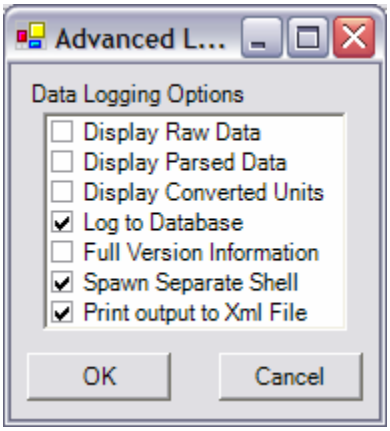


Figure 5-5. Advanced Logging Dialog Window

If you are not able to receive data, you should check to see if your nodes are transmitting by viewing the data logging screen at the bottom of your *MoteView* display. Depending upon the selections you made in the Data Logging window, you may see data in raw, converted or parsed format.

Table 5-1. Description of advanced logging options and the equivalent *XServe* flag

Advanced Logging Option	Description	*XServe* flag
Display Raw Data	The raw packet bytes coming into the PC.	**-r**
Display Parsed Data	The raw ADC values displayed as parsed fields.	**-p**
Display Converted Data	The incoming data converted to engineering units.	**-c**
Log to Database	Will store raw ADC values to *PostgreSQL* database.	**-l**
Full Version Information	Display exact version for all board modules	**-v**
Spawn Separate Shell	Launches *XServe* in command line shell	n/a

The Start button will be enabled if *XServe* is not already running. Clicking on **Start** will start *XServe* running as a separate process with the given options. The output of *XServe* will be streamed to the **Server Messages** pane unless **Spawn Separate Shell** has been selected in which case a *Command Prompt* window, as shown above in Figure 5-6, will be created. Sample outputs from various Crossbow applications are listed below.

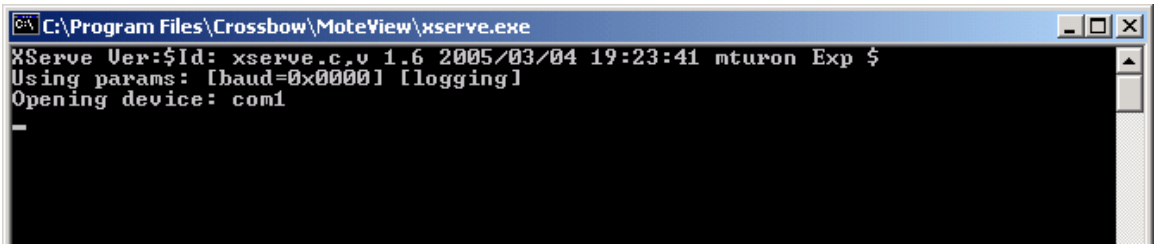

C:\Program Files\Crossbow\MoteView\xserve.exe
XServe Ver:$Id: xserve.c,v 1.6 2005/03/04 19:23:41 mturon Exp $
Using params: [baud=0x0000] [logging]
Opening device: com1

Figure 5-6. Portion of a *Command Prompt* window that appears once the data logging has started

XMTS101 Logging Output

```
INSERT into mts101_results (result_time,nodeid,parent,voltage,temp,light)
values (now(),0,126,377,183,186)
```

XMTS310 Logging Output

```
INSERT into mts310_results
(result_time,nodeid,parent,voltage,temp,light,accel_x

,accel_y,mag_x,mag_y,mic) values
(now(),0,126,378,197,301,219,193,177,210,215)
```

XMTS400 / XMTS420 Logging Output

```
INSERT into mts420_results
(result_time,nodeid,parent,voltage,humid,humtemp,inte

rsemacal1,intersemacal2,intersemacal3,intersemacal4,prtemp,press) values
(now(),

0,126,377,1391,6517,45446,46808,38941,45989,26835,18130)
```

XMEP410 Logging Output

```
INSERT into enviro_results
(result_time,nodeid,parent,epoch,voltage,humid,humtem
p,inthum,inttemp,accel_x,accel_y,photo1,photo2,photo3,photo4,prtemp,press)
values (now(),0,126,0,379,0,0,65535,65535,201,0,174,200,207,0,65535,65535)
```

XMTS510 Logging Output

```
INSERT into mts510_results
(result_time,nodeid,parent,light,accel_x,accel_y,mic)
 values (now(),0,126,273,222,205,245)
```

XMDA300 Logging Output

```
INSERT into mda300_results
(result_time,nodeid,parent,voltage,humid,humtemp,adc0
,adc1,adc2,digi0,digi1,digi2) values
(now(),0,126,377,960,6678,2500,2243,2505,0,0,0)
```

XMDA500 Logging Output

```
INSERT into mda500_results (result_time,nodeid,parent,voltage,temp,adc2,adc3,
adc4, adc5, adc6, adc7) values (now(),0,126,191,527,306,240,204,213,191,234)
```

XMEP510 Logging Output

```
INSERT into enviro_results
(result_time,nodeid,parent,epoch,voltage,therm,humid, humtemp) values
(now(),0,126,15,376,517,65535,65535)
```

5.2.6 MoteConfig Programming Utility

MoteConfig is a GUI utility for programming Motes by providing an interface for downloading pre-compiled *XMesh*/TinyOS firmware applications. A major benefit of *MoteConfig* is that Mote platforms can be programmed without having to install the TinyOS programming environment.

1. MoteConfig dialog window can be started by clicking on **Tools > Program Mote...** This will bring up a window shown in Figure 5-7. Click **Local Program** on tab.

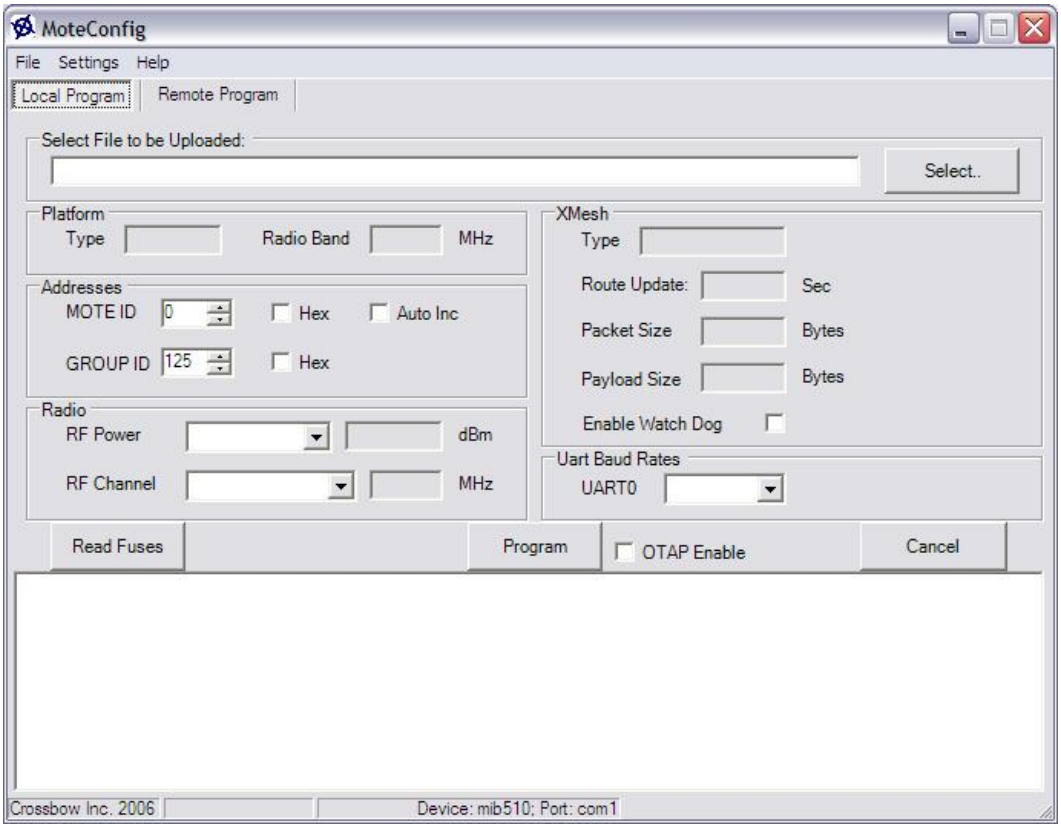
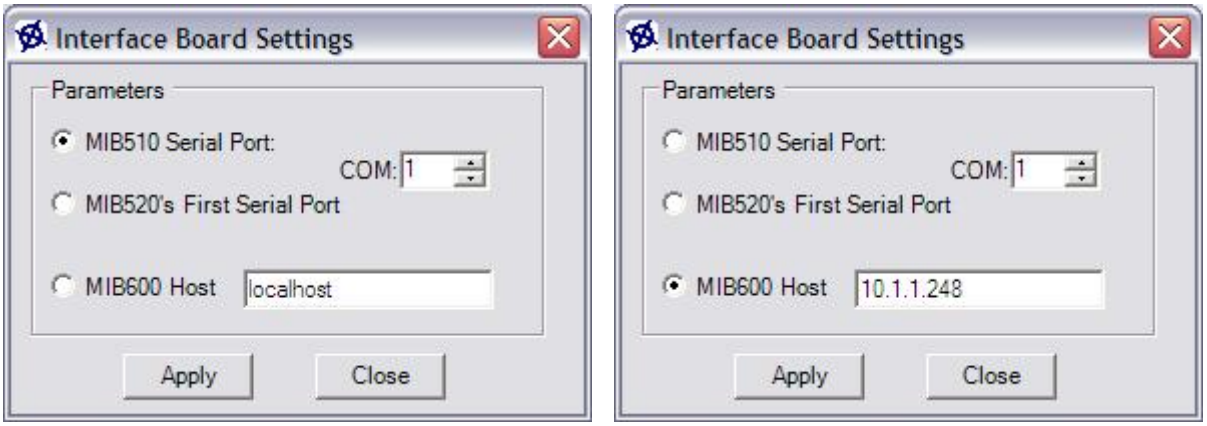


Figure 5-7. Screenshot of MoteConfig GUI Window

2. Click on **Settings > Interface Board Settings...** and choose the proper MIB gateway and check that the port settings are correct.

☑ **EXAMPLE 5-1:** Left screenshot shows **Interface Board Settings** for an MIB510 on COM port 1. Right screenshot shows **Interface Board Settings** for an MIB600 with IP address 10.1.1.248 and on the same LAN as that of the PC.



---

3. Right click on the **Select…** button on the top right which will open up a directory window (see Figure 5-8). Or navigate to another directory which has the pre-compiled Mote application you wish to install. Select the specific **.exe** file of the application you want to download into the Mote.

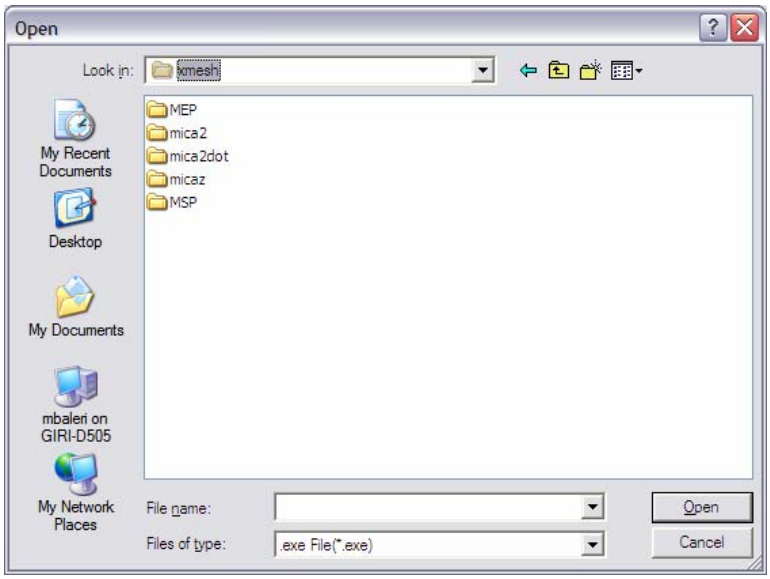


Figure 5-8. Screenshot for selecting XMesh applications

NOTE: The XMesh applications distributed with MoteView now require *XMeshBase_xxx.exe* programmed with MOTE ID=0 as base station application.

The binary scan feature built into MoteConfig will display the default parameters programmed into this application.

4. Specify desired **MOTE ID, Group ID, RF Power,** and **RF Channel** from the drop down menus.
5. Right-click on the **Program** button and you will see the resulting output at the bottom of the screen (see Figure 5-9).

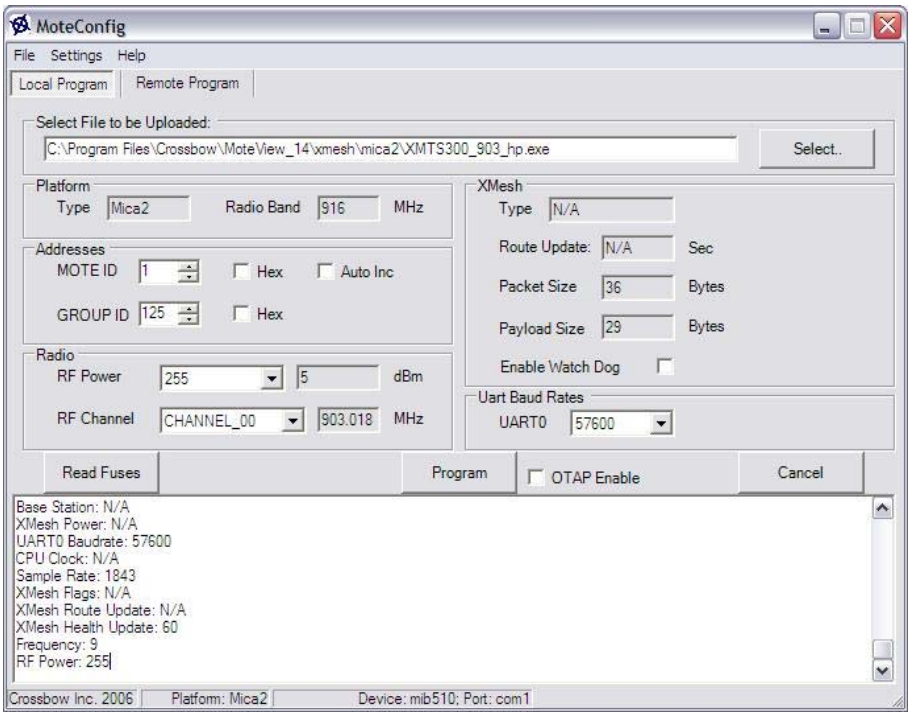


Figure 5-9. Screenshot of MoteConfig which has successfully installed the XMTS300 application on a MICA2 Mote set to a node ID of 1, group ID 125, radio channel 0 (f = 903.018 MHz), and at 5 dBm power (255)

The Table 5-2 below describes the advanced options available in *MoteConfig* GUI.

Table 5-2. Description of advanced logging options and the equivalent *XServe* flag

Advanced Options	Description
Hex	This checkbox lets users specify the ID as hexadecimal value.
Auto Inc	This checkbox allows users to automatically increment MOTE ID field.
Enable Watch Dog	This checkbox will enable a watchdog timer. *XMesh* uses a watchdog component to perform a watchdog reset after it has missed 5 route update messages in a row.
UART0	Allows the user to change the default UART baud rate of the Mote
OTAP Enable	Allows users program Mote with OTAP Image (refer to section 8.1)

NOTE: The MoteConfig utility can also be used to program the Motes over the air using *XOtap* feature of *XMesh*. The step-by-step procedure on how to do this will be explained in Chapter 8 later.

The MoteConfig also allows users to overwrite default fuse settings of ATmega128 processor. This dialog window can be started from **Settings>Fuse Settings** as shown in Figure 5-10. Checking on **Override default fuse settings** allows you to change the other default settings.

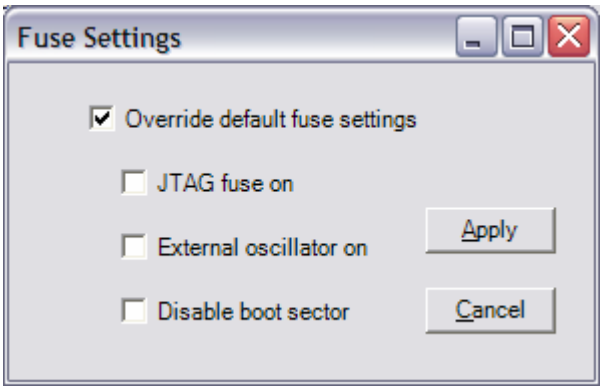


Figure 5-10. Fuse Settings Dialog box in MoteConfig

Available options are:

- **JTAG fuse on** – activates JTAG debug mode for ATmega 128 processor. When enabled, the processor draws ~3 mA and hence should be disabled for Low Power apps. By default, JTAG fuse is turned off for all the *XMesh* apps.
- **External oscillator on** – forces *XMesh* apps to use external oscillator for timer. This draws more current than internal oscillator. By default, Low Power apps have this switch disabled and High Power apps have this enabled.
- **Disable boot sector** – makes the Mote not execute the boot loader section and directly start execute the mail program. This should not be checked if you are using the OTAP functionality of XMesh apps. By default, boot sector is enabled.

5.2.7 Alerts

MoteView alert manager allows users to define alert condition based on any sensor data of any sensor node. An Alert is a user programmable event that gets triggered when sensor data exceeds pre-defined threshold. An Alert is composed of several pieces:

1. **Alert ID** – A sequential number to identify the alert
2. **Node Name** – the node whose conditions can trigger an alert
3. **Sensor Name** – the sensor whose readings can trigger an alert
4. **Alert Condition** – the comparison operation to decide whether an alert has triggered
   - Equals
   - Exceeds
   - Below
   - Exceeds or Equals
   - Below or Equals
5. **Alert Threshold** – the value to compare the alert condition against
6. **Unit** – the unit for the alert threshold
7. **Alert Action** – the operation to perform in response to a triggered Alert

   There are two possible responses to an Alert:

   - Display a message dialog and console message
   - Send an email (allows text paging a mobile phone)
8. **Alert Interval** – specify the interval for the action to be taken.

To set an Alert, follow these steps below:

1. Chose from **Tools > Alerts > Alert Manager...** and click the button "**Add New Alert Item**". This operation will add a blank alert item to the list with Alert ID 1 as shown in Figure 5-11.

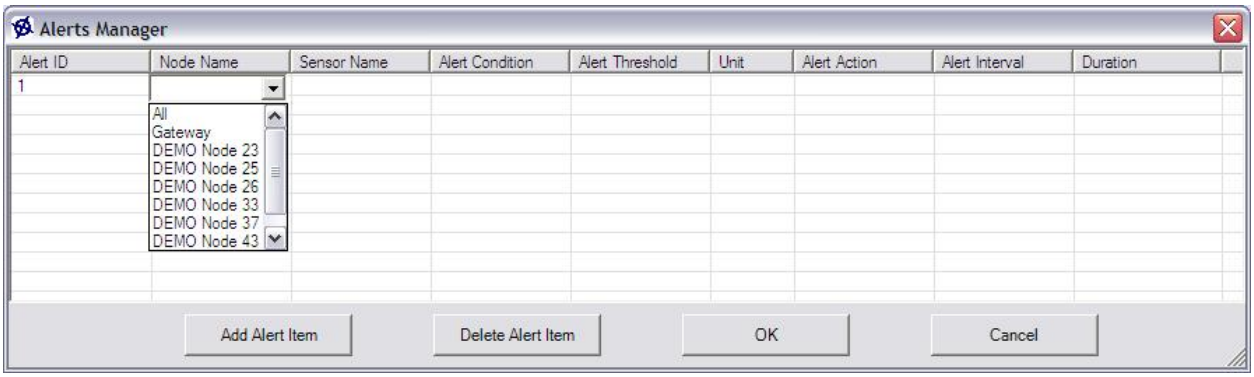


Figure 5-11. Screenshot of the Alert Manager

2. From the **Node Name** drop-down list, select a node that is currently a part of your deployment.
3. From the **Sensor Name** drop-down list, select a sensor that you want to use to trigger the Alert. This automatically updates the Unit column associated with that particular sensor type.
4. From **Alert Condition** drop-down list, specify an appropriate condition to trigger the Alert, viz. "**<**", "**>**", "**<=**", "**>=**", "**=**".
5. For **Alert Threshold** field, input numerical value in the text box.
6. From **Alert Action** drop-down list, chose an action, "**Send Alert Email**" or "**Pop-up Alert Form**".
7. From the **Alert Interval** drop-down list, specify the interval for the Alert to be sent.
8. From **Duration** drop-down list, specify the time duration for which the Alert Condition needs to be met before the Alert Action can be triggered.

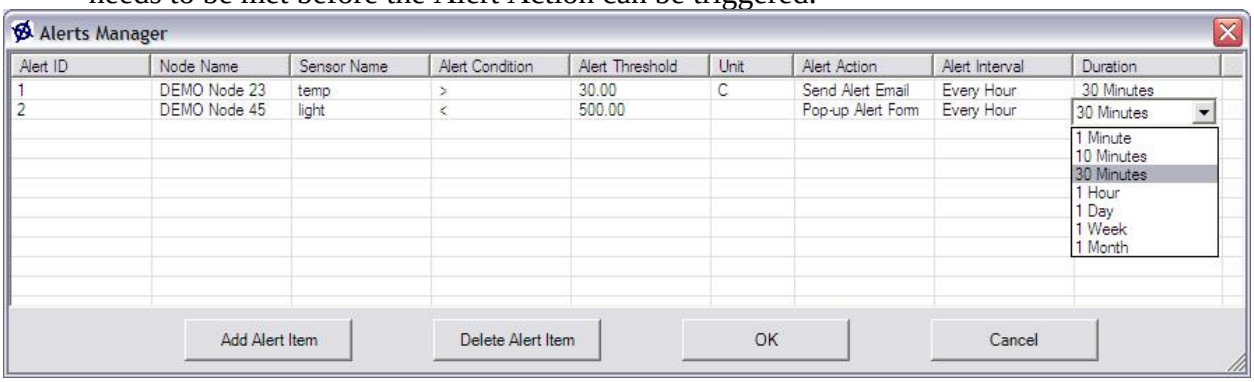


Figure 5-12. Screenshot of the Alert Manager after adding two Alerts

Deleting an Alert is very straight forward, first select and highlight the item that you wish to delete then click the "**Delete Alert Item**" button, the selected alert item will be deleted.

For the Alert manager to send Email, the users should first configure the mail settings. This can be accomplished as follows:

1. Click from **Tools > Alerts > Alert Mail Configure...** and that will bring up the dialog box shown in Figure 5-13
2. Enter the **SMTP Server** name of your mail server
3. Specify the **User Name** and **Password** of your mail account
4. In **Send To** field, specify the Email address where the alert needs to be sent
5. Enter any optional **Message** to be sent in the body of the Email
6. Click on **OK.**

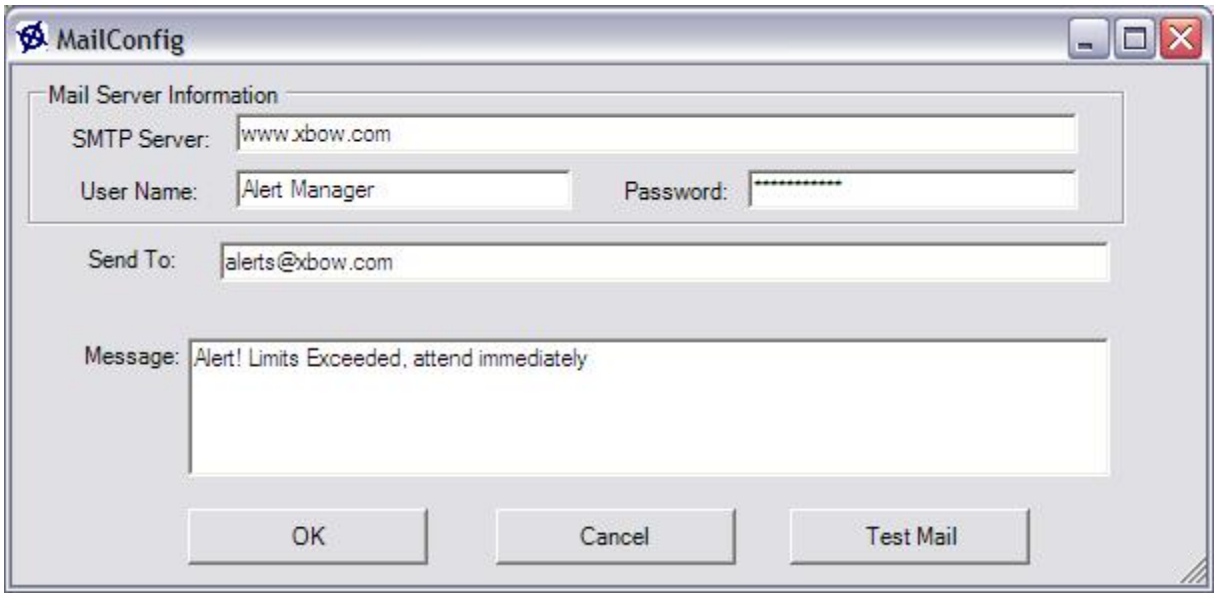


Figure 5-13. Screenshot of the Alert Mail Configuration

5.2.8 MDA300 Calibration Support

MoteView allows users to define their own calibration parameters to convert from raw voltage into engineering units for external sensors when connected via MDA series data acquisition boards. Users can specify the calibration coefficients for each of the available channels.

MDA Support interface allows the users to define custom external sensors, conversions, and associate them with channels on a data acquisition board. It is composed of several pieces:

- MDA Board Type – MDA100, MDA300, MDA320 or MDA325, MDA500
- Sensor Type – Configuration name associated with an ADC channel
- Configuration Name – assigned to a particular sensor
- Units – Engineering units associated with a particular configuration/conversion
- Conversion type – Up to third order polynomial or look up table

To create a new external sensor configuration, follow these steps:

1. MDA Configuration dialog window can be started by clicking on **Tools > MDA Support.** This will bring up a window shown in Figure 5-14**.**

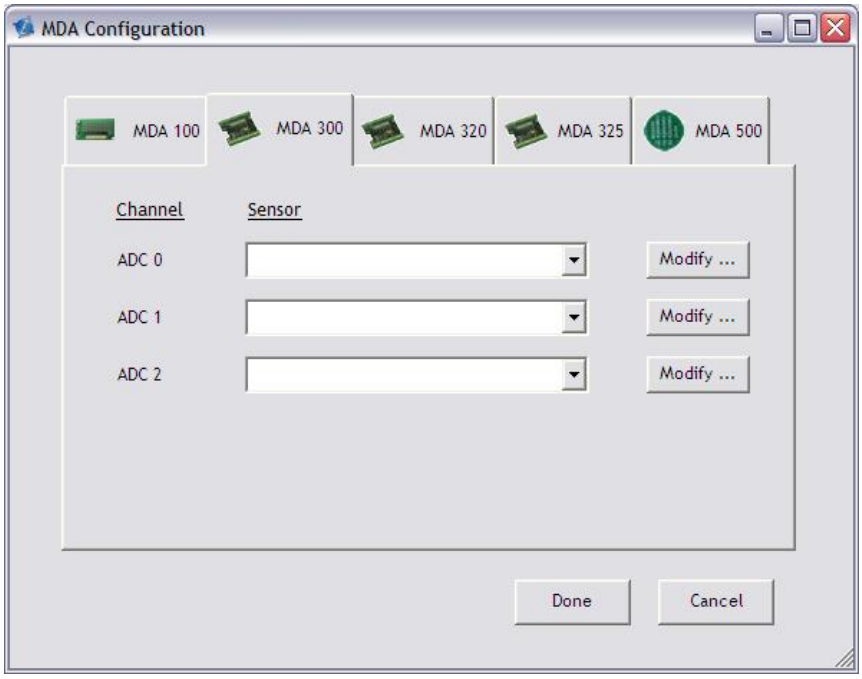


Figure 5-14. Screenshot of the MDA Configuration Support

NOTE: You need to be connected to mdaxxx_results table in the data base before you can use the **MDA Support** dialog.

2. Select the specific MDA board type that you are currently using to connect external sensors.
3. Next to the Channel of interest, from the **Sensor** dropdown box see if any of the choices is relevant to you. If not, click on **Modify...** This will bring up a dialog window for Sensor Configuration as shown in Figure 5-15.

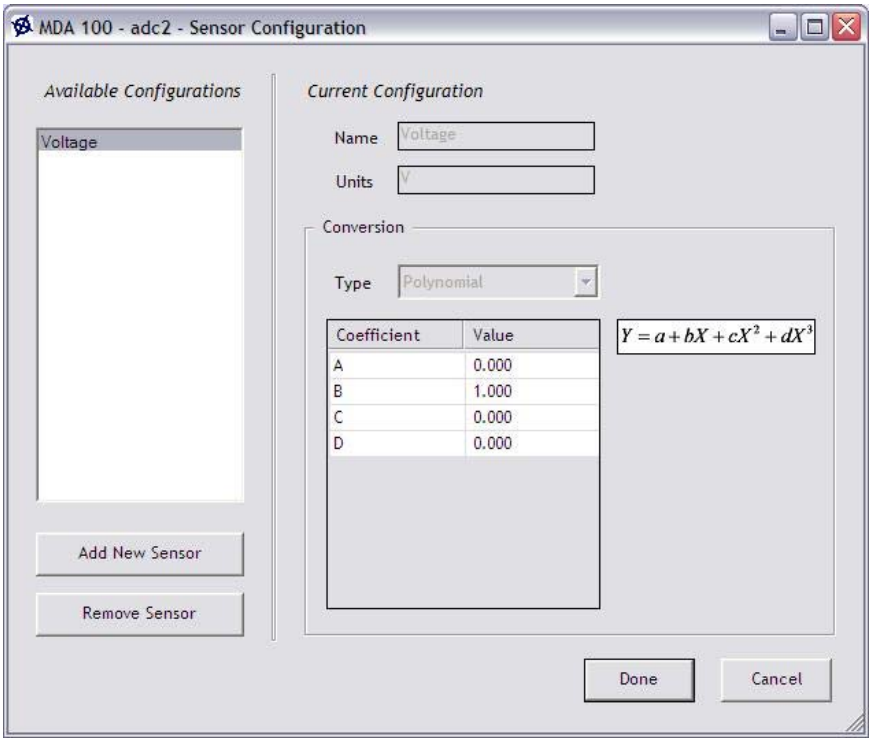


Figure 5-15. MDA Sensor Configuration

4. Click on **Add New Sensor**.
5. Edit the **Name** and **Units** under Current Configuration group box.
6. Under Conversion group box from the **Type** dropdown, select the calibration type (i.e. **Polynomial**, **Look Up Table** or **Thermistor**).
    a. If you used **Polynomial** as the conversion type, then specify the Coefficient values (a, b, c and d) to convert from raw voltage (X) into engineering units (Y) as shown in Figure 5-16.

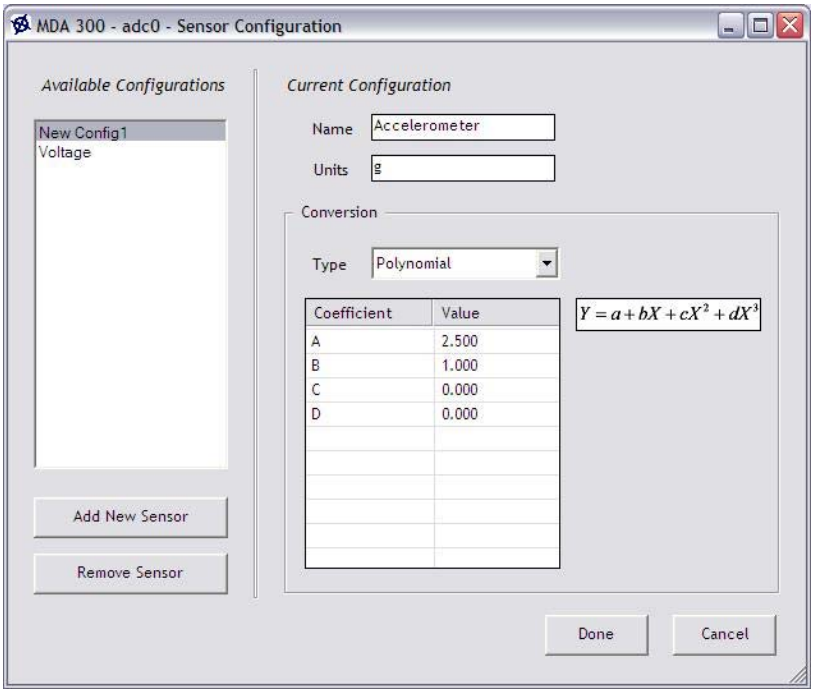


Figure 5-16. MDA300 Sensor Configuration using the Polynomial Conversion

b. If you used **Look Up Table** as the conversion type, then input the **Voltage** readings and corresponding engineering values in the table as shown in Figure 5-17. Users can add or delete a row using the **Add Row** or **Delete Row** buttons.

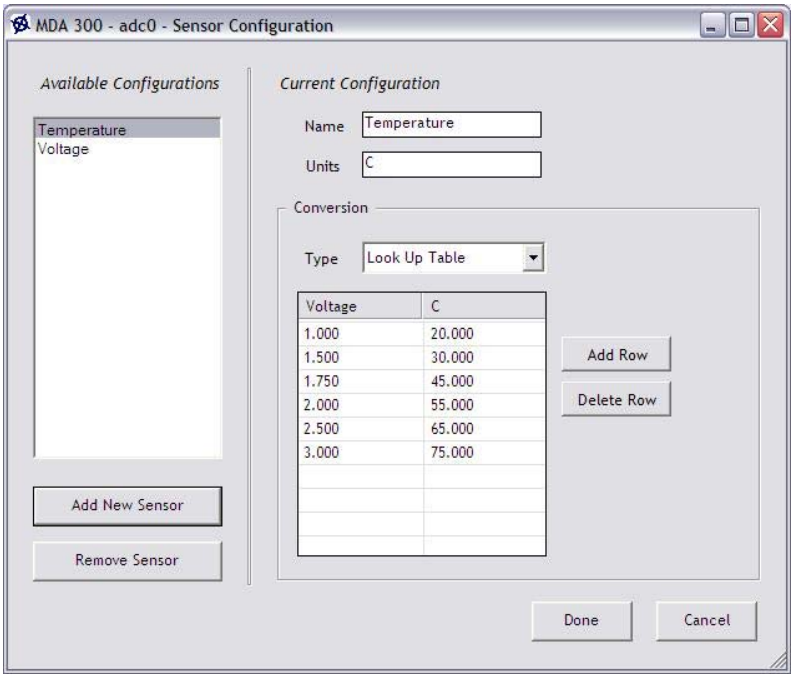


Figure 5-17. MDA300 Sensor Configuration using the Look Up Table Conversion

a. If you used **Thermistor** as the conversion type, input the excitation voltage, external fixed resistance and the Steinhart-Hart coefficients as shown in Figure 5-18. Note that the converted temperature is only available in degrees C.

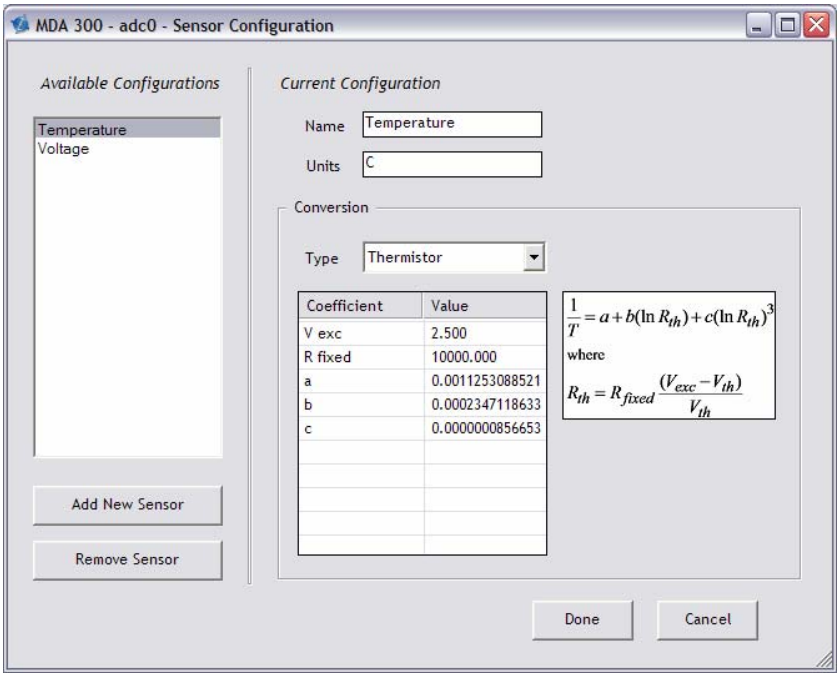


Figure 5-18. MDA300 Sensor Configuration using the Thermistor Conversion

7. Click on **Done** for the new Sensor Configuration to take effect.
8. This saved Sensor Configuration will now be available in **Sensor** dropdown box of Figure 5-14 and can then also be used for other ADC channels.
9. Users can remove a particular sensor configuration by highlighting it and then clicking on **Remove Sensor**.

5.3 Units

The units menu allows the user to select their preferred engineering units for a given class of sensor. All sensors of that class will convert to the units specified by the user. A user may select to view raw data as direct ADC readings instead of a particular unit.

5.3.1 Temperature

MoteView supports temperature conversion to Celsius (C), Fahrenheit (F), and Kelvin (K).

5.3.2 Pressure

MoteView supports pressure conversion to Atmosphere (atm), Bar (mbar), Pascal (Pa), Per mm of Hg (torr), and Pounds per square inch (psi).

5.3.3 Acceleration

MoteView supports acceleration conversion to Meters per second squared (m/sec^2), and Relative gravity (g).

5.4 Windows

5.4.1 Server Configuration

This menu will bring up the Server Configuration dialog.

5.4.2 Data

This menu will bring up Data View tab to the front of the display.

5.4.3 Charts

This menu will bring up Chart View tab to the front of the display.

5.4.4 Topology

This menu will bring up Topology View tab to the front of the display.

5.5 Help

5.5.1 MoteView Help

This menu will bring up this document, *MoteView User's Manual*.

5.5.2 Tutorial

This menu will bring up a tutorial document for *MoteView*.

5.5.3 Support

This menu will open a browser window to the Crossbow support webpage if your PC is connected to the internet.

5.5.4 About

This menu will bring the about screen and display the exact *MoteView* version number.

5.6 Toolbar Menu

The toolbar provides quick access to some of the more common features of the *MoteView* application. From left to right, the toolbar provides buttons for database connection, data

logging with *XServe*, *MoteConfig*, Alerts Manager, MoteView settings, export to CSV, print, and print preview, refreshing the view, and displaying live (vs. historical) sensor network data.

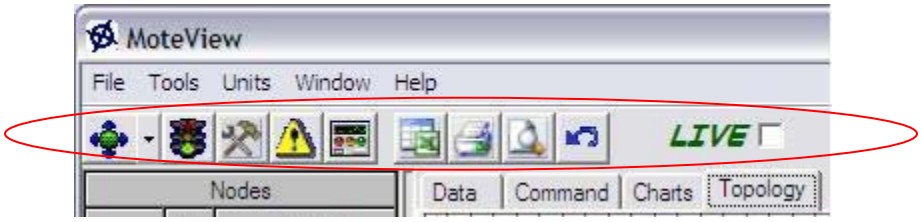


Figure 5-19. Screenshot of toolbar menu of the MoteView

5.7 Dialogs

5.7.1 Database Server Configuration

Server configuration is a critical first step to using *MoteView*. The Database Server Configuration dialog can be opened in three ways:

1. Click the Connection icon on the Tool bar
2. Select **File > Connect...**
3. Select **Window > Server Configuration...**

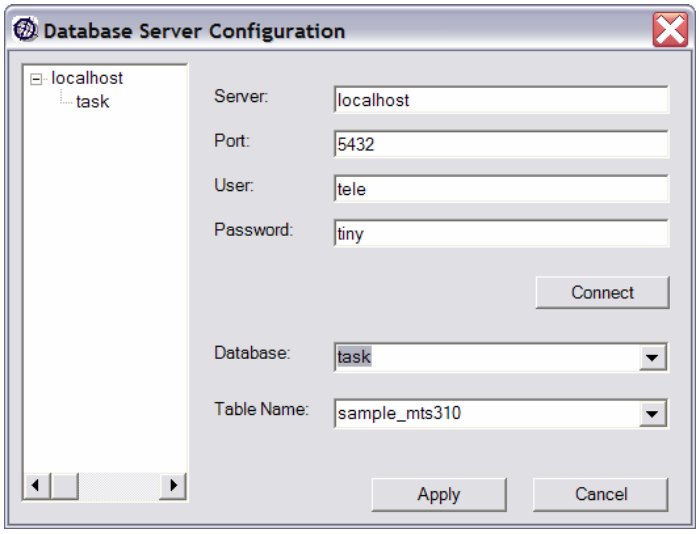


Figure 5-20. The Database Server Configuration dialog box

From within the **Database Server Configuration** dialog window, you can enter the following fields:

- **Server**: Can be a hostname or IP address. To specify your workstation as a local server enter **`localhost`**.
- **Port**: TCP/IP port of *PostgreSQL* database server. Defaults to **`5432`** which is the standard *PostgreSQL* port.
- **User:** username for database server - **`tele`** is the default.

- **Password**: password for database server - **`tiny`** is the default.

After typing in the four fields above, click the **Connect** button to attempt to connect to the said database and update the pull down lists for what is available on that server.

To complete server selection, the pull down lists should then be used to specify which result set on the server is to be viewed:

- **Database**: All available *PostgreSQL* databases on the server will be displayed
- **Table Name**: The table files containing results to be viewed. This is populated with available tables.

After selecting a database, table, and client, the user may click **Apply** to have *MoteView* display the results data for the selections. **Cancel** will close the dialog window.

Saved configurations are stored in the tree view to the left in a nested fashion. The server names are listed with the database names enclosed within. Clicking a server name opens the list of saved databases on that server. Clicking on a database name loads those settings into the fields on the right. To connect to a saved configuration, click on the database name and click **Apply**. When a database has multiple result tables, the last saved table will be used.

5.7.2 Mote Properties

This dialog is opened by selecting the **Properties** menu when right clicking a node in the Topology map, or by double-clicking a node in the node list. It allows the user to assign a **Name** and **Group** of the Mote.

The **Color** tab allows the user to select the charting color for the Mote.

NOTE: The **Sensorboard** tab of this dialog is only used to display calibration coefficients for boards that are equipped with an Intersema pressure temperature sensor such as the MTS400, MTS420, and MEP410.

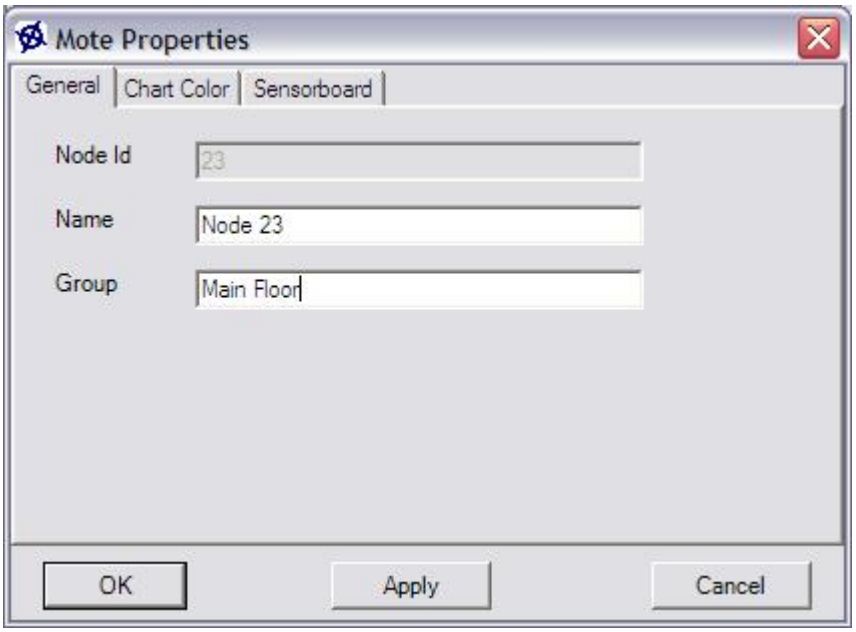


Figure 5-21. Screenshot of the Mote Properties dialog window

This dialog is opened by selecting the **Properties** menu when right clicking a node in the **Topology** map, or by double-clicking a node in the node list.

5.8 Server Messages

The S**erver Messages** pane is the bottom section of the *MoteView* window and displays server side messages, database errors, and general status messages as *MoteView* is running.

6 Server Administration

XServe can be used as a data logging server and comes preinstalled with *MoteView*. A user may run *XServe* from a *Cygwin* command line (included in the TinyOS 1.1.0 installer), or directly within *MoteView* using the data logging dialog accessible from the **Tools > Log Data** menu.

NOTE: To use the command line interface for *XServe* you **must** have the *Cygwin* shell installed on your PC. Instructions for installing *Cygwin* from *MoteWorks* InstallShield Wizard are detailed in the *MoteWorks Getting Started Guide*.

6.1 General configuration

Generally, the installation wizard will install *XServe* automatically. The `cygwin1.dll` library in the *MoteView* installation directory must exactly match the `cygwin1.dll` library in the user's *cygwin/bin/* directory for proper operation.

6.2 XServe Logging

XServe is a general purpose command line tool for displaying sensor readings being received by a gateway Mote. If using a *Cygwin*, type **`xserve -?`** to display the command line options.

```
Usage: xserve <-?|r|a|p|c|xr|xp|xc|dbxmlr|xmlp|xmlc|v|alert|m>
              <-l=tablename>
              <-dbserver=servername> <-dbport=portnum>
              <-dbname=database name> <-dbuser=username> <-dbpasswd=password>
              <-h=path,hostname,portnum,config_file>
              <-m=com,baud,protocol,slaveaddress,defaultregistervaluesas>
              <-xmlfile=filename> <-xmlport=portnum>
              [<-sf=hostname:port> | <-fsf=hostname:port> | <-device=dev>]
              <-port=num> <-baud=num> <-platform=plt>
              <-debug=level>
              <-configfiles=filename:filename:>
              <-loadparsers=filename:filename:...>
              <-loaddatasinks=filename:filename:...>
              <-heartbeat=<num missed>

   -?         = display help [help]
   -r         = raw display of tos packets [raw]
   -a         = ascii display of tos packets [ascii]
   -p         = parsed display of tos packets [parsed]
   -c         = converted display of tos packets [conveted]
   -xr        = raw tos packets xported to file [export raw]
   -xp        = parsed tos packets exported to file [export parsed]
   -xc        = converted tos packets exported to file [export converted]
   -db        = parsed tos packets exported to db [database parsed]
   -dbserver  = database server name (default=localhost)
   -dbport    = database server port number (default=5432)
   -dbname    = database name (default=MoteView db)
   -dbuser    = database user (default=MoteView user)
   -dbpasswd  = database user password (default=MoteView user password)
```

```
   -l         = parsed tos packets exported to db
                (deprecated) [database parsed]
   -xmlr      = raw tos packets exported to xml [xml raw]
   -xmlp      = parsed tos packets exported to xml [xml parsed]
   -xmlc      = converted tos packets exported to xml [xml converted]
   -xmlfile   = file name to store exported xml (default=screen)
   -xmlport   = port number to start the xml server
   -v         = show version of all modules
   -h         = display data through web server
   -m         = export data using modbus
   -port      = set server port <default = 9001>
   -sf        = connect to unframed serial forwarder
   -fsf       = connect to framed serial forwarder
   -device    = connect to serial device <default = /dev/ttyS0>
   -baud      = set serial baud rate <default = 57600>
   -platform  = set platform. <default = mica2>
                values=mica2dot|mica2|mica|telos|micaz
   -debug     = set debug level. <default = DBG_WARNING>
   -alert     = alert when data values are above/below specified ranges
   -daemon    = run in daemon mode
   -nomonitor = run without a system monitor
   -configfiles      = load xml configuration files.
   -loadparsers      = load only the listed parsers files from the
                       shared_lib. (default=all files are loaded)
   -loaddatasinks    = load only the listed datasinks files from the
                       shared_lib. (default=all files are loaded)
   -heartbeat        = turn on the heartbeat monitor and reset after <num
                       missed>
   -convZto2         = convert incoming network packets from micaZ headers to
                       mica2 headers and vice versa
   -conv2toZ         = convert incoming network packets from mica2 headers
                       to micaZ headers and vice versa
   -joinfile         = node mapping file for unique ids joining the nextwork
   -noorganic        = disable organic growth for joining nodes
```

XServe will be installed automatically by the *MoteView* Setup installation procedure. Data logging to a database was added to *XServe* in version 1.11. To start *XServe* data logging, from a *Cygwin* command line type in a command similar to the following. Be sure to specify the proper input device (**`-i`**) and port.

Table 6-1. Description of *XServe* Logging Options

***XServe* Command Line for Data Logging**	**Description**
`xserve -l`	Log readings from a MICA2 as gateway on COM1 (default)
`xserve –l –s=com4`	Log readings from a MICA2 as gateway on COM4
`xserve -l -b=mica2dot`	Log readings from a MICA2DOT as gateway on COM1 (default)
`xserve –l –r –c –i`	Log data to database, display raw packets, and display converted readings from *SerialForwarder* on localhost port 9001

NOTE: Every time you reboot your computer and want to keep logging data, you must open a *Cygwin* window and type **`xserve -l`** to start logging data.

7 Database Administration

7.1 PostgreSQL

> **NOTE:** To use the command line interface for *PostgreSQL* you **must** have the *Cygwin* shell installed on your PC. Instructions for installing *Cygwin* from *MoteWorks* InstallShield Wizard are detailed in the *MoteWorks Getting Started Guide*.

PostgreSQL is an advanced relational database system that is provided with the *Cygwin* on the PC and is available on the Stargate. The database tables that *MoteView* accesses can be manipulated directly by advanced users. To access the *PostgreSQL* database, from a *Cygwin* shell type

```
psql –h localhost –U tele task
```

Below is an example of what you should get as a response to that command:

```
$ psql -h localhost -U tele task
Welcome to psql 7.4.5, the PostgreSQL interactive terminal.


Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit


task=#
```

7.2 SQL

SQL is the generic command language used to manipulate databases such as *PostgreSQL*. SQL commands can be typed in directly from the *PostgreSQL* command shell. A list of common and useful commands follows:

7.2.1 Display all readings

Type

```
select * from <tablename>;
```

The **`select`** statement will display results out **`from`** the given **`<tablename>`**. The **`*`** character is a wildcard meaning that all columns should be displayed.

```
task=# select * from surge_results;
        result_time         | epoch | nodeid | parent | light | temp | voltage |
mag_x | mag_y | accel_x | accel_y
----------------------------+-------+--------+--------+-------+------+---------
+-------+-------+---------+---------
 2004-07-12 12:47:54.29425  |   757 |      4 |      0 |   182 |  135 |     229 |
255 |   255 |     200 |     200
 2004-07-12 12:48:02.26325  |   758 |      4 |      0 |   181 |  135 |     229 |
252 |   252 |     200 |     200
```

7.2.2 Display subset of readings

Type

select *field*$_1$, *field*$_2$,... from <tablename> where *field*$_n$ condition *value*;

The **select** statement will display results out **from** the given **<*tablename*>**. The * character is a wildcard meaning that all columns should be displayed.

```
task=# select result_time,temp,light from surge_results where nodeid=1;
        result_time         | temp | light
----------------------------+------+-------
 2004-07-12 12:48:07.31025  |  136 |   234
 2004-07-12 12:48:15.29325  |  136 |   232
```

7.2.3 Rename a table

Type

ALTER TABLE <*tablename*> RENAME TO <*newname*>;

7.2.4 Delete all readings from table

Type

DELETE FROM <*tablename*>;

7.2.5 Deleting specific readings from table

To delete all results before the specified date, type

DELETE FROM <tablename> WHERE result_time < '2004-11-20';

To delete all results with ADC voltage reading greater than 400.

DELETE FROM <tablename> WHERE voltage > 400;

To delete all results from node number 3.

DELETE FROM <tablename> WHERE nodeid = 3;

7.2.6 Delete table entirely

Type

DROP TABLE <tablename>;

7.3 Database Tools

PostgreSQL comes with other tools for offline manipulation of data besides the psql shell. The more useful of these are described here. These windows command prompt version of these tools are installed to *C:\Program Files\PostgreSQL\8.0.0-rc1\bin* by default. The *psql* tool is available from the *Cygwin* command prompt as well.

7.3.1 PostgreSQL Export

To output entire task database to a file, e.g., `my_database.out`.

```
pg_dump –h localhost –U tele –f my_database.out task
```

To save contents of `surge_results` table to a file of SQL commands named `surge.out`:

```
pg_dump –h localhost –U tele –t surge_results –f surge.out task
```

7.3.2 PostgreSQL Import

To load files from a PostgreSQL exported table, use the following command:

```
psql task < surge.out
```

8 Over-the-Air-Programming (OTAP) using MoteConfig

The Over-the-Air-Programming (OTAP) feature allows users to reprogram any Mote within the *XMesh* network. OTAP allows one or more Motes to receive new programming images from *XServe* (via *XOtap*, a server-side application via wireless communication.

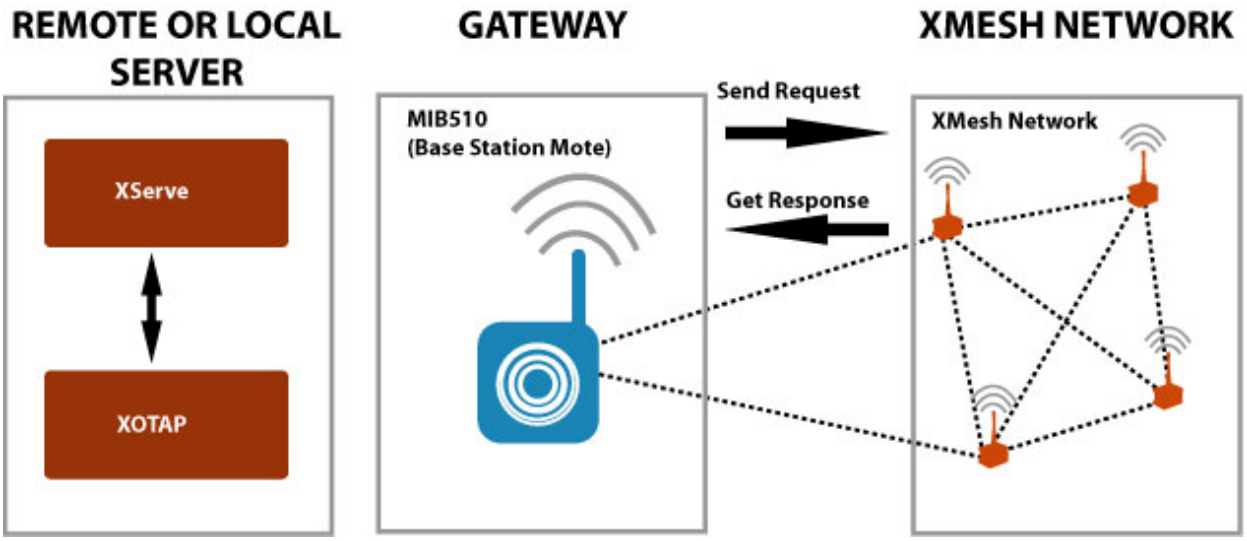


Figure 8-1. *XOtap* Architecture

8.1 OTAP Preparation using *MoteConfig* Utility

Before you can program the Motes in the network over the air, you need to prepare them with bootloader and OTAP image. You can use MoteConfig utility to accomplish this. Follow these steps to prepare the Motes for over the air programming.

1. Start the *MoteConfig* from **Start>Programs>Crossbow>MoteConfig**
2. From **Settings>Interface Board Settings...** select appropriate programming interface board and port

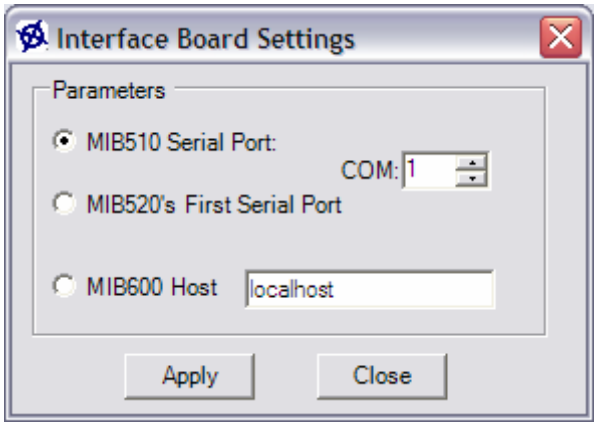


Figure 8-2. Programming Interface Board Settings in *MoteConfig*

3. OTAP enabled programming using UISP – Click on the **Local Program** tab and click on **Select**. Browse to an *XMesh* application file (main.exe). Choose appropriate **Mote ID**, **Group ID**, **RF Power**, **RF Channel** and make sure that **OTAP Enable** box is checked. Click on **Program**.

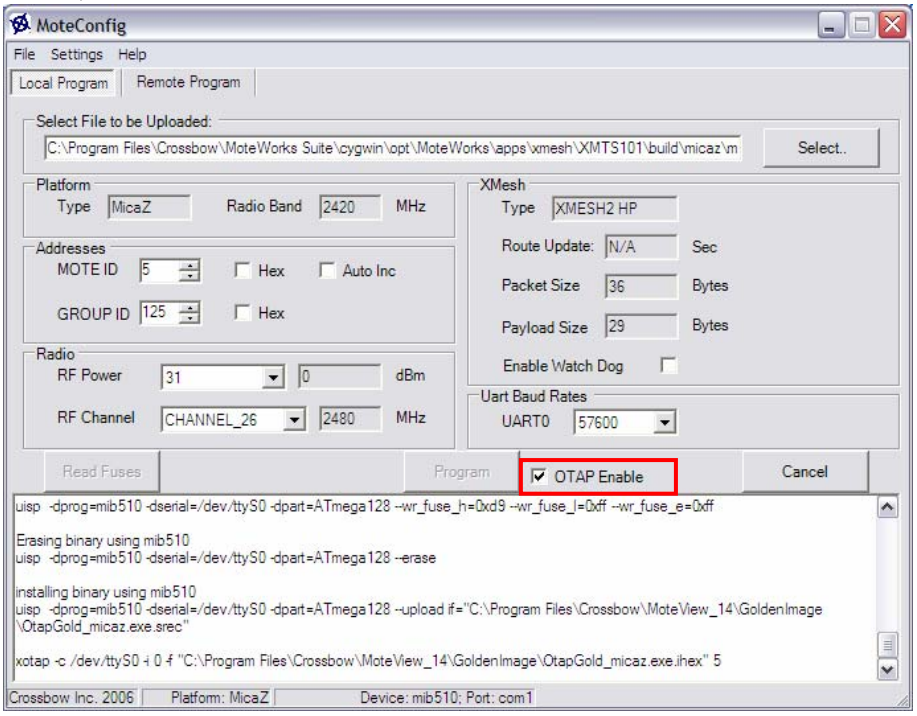


Figure 8-3. Programming *XMTS101* OTAP-enabled application using *MoteConfig*

4. Repeat the Step 3 for all the nodes in the network. When the bootloader is installed successfully, the LEDs count up twice when the node is switched on.
5. Program base station Mote with *XMeshBase* application, Mote ID = 0 and **OTAP Enable** unchecked.

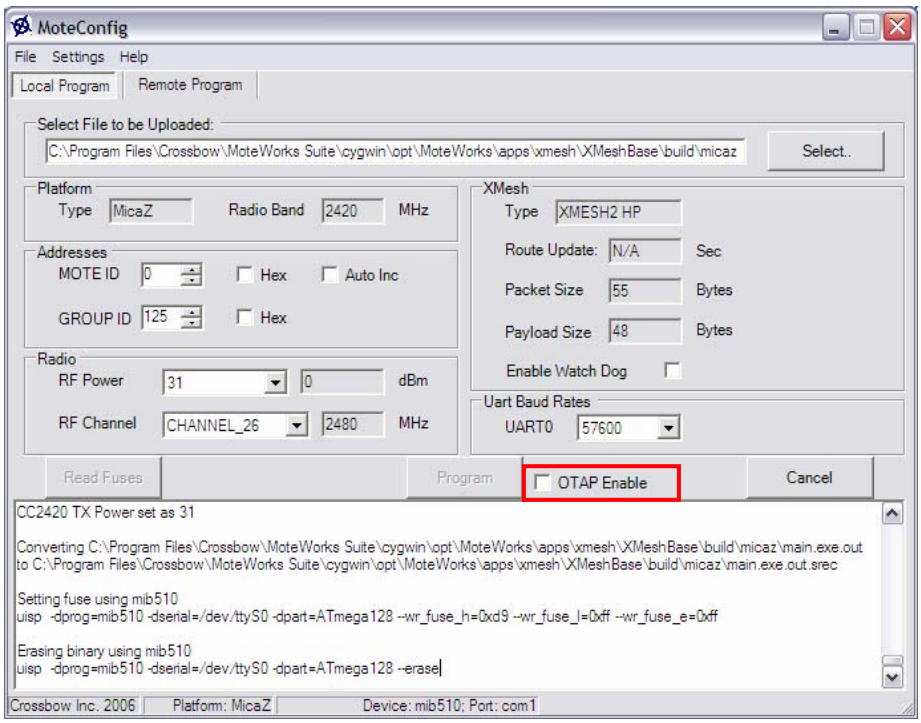


Figure 8-4. Programming *XMeshBase* application into base station using *MoteConfig*

8.2 OTAP using *MoteConfig* Utility

Once all the Motes are prepared for OTAP, you can then program them over the air. The following steps explain how to use MoteConfig to program the nodes in the network.

1. Connect the base station to the PC interface board and turn on the remote nodes that were prepared as per Section 8.1.
2. **Listen** – If you are not sure which nodes are part of your network, you can listen to the network to get list of nodes. Simply click on **Listen** button of the *MoteConfig* GUI and wait for sometime. This automatically starts up the *XServe* and the node IDs discovered in the network will be displayed on the left hand side as shown in Figure 8-5.

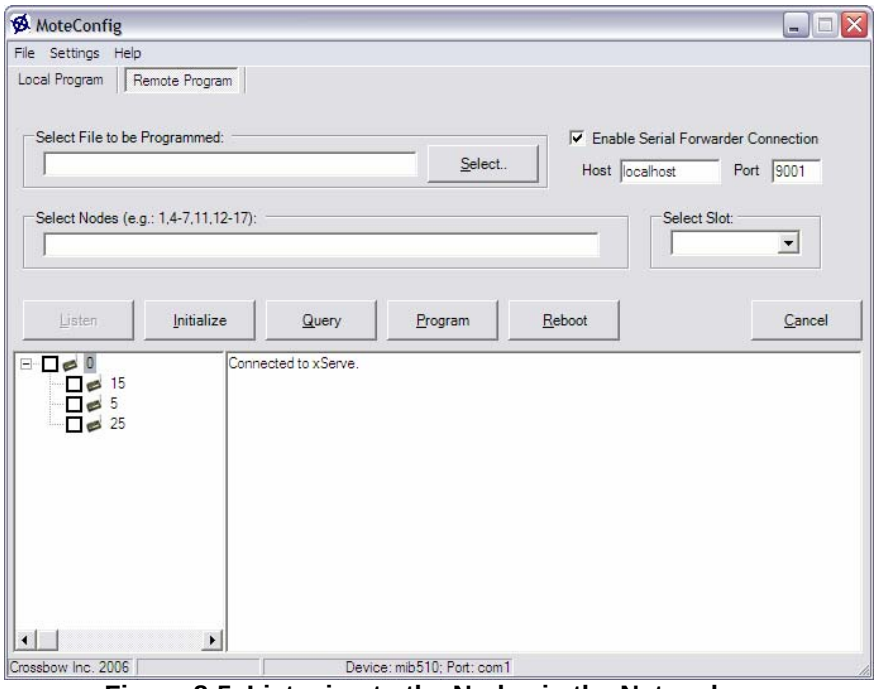


Figure 8-5. Listening to the Nodes in the Network

3. **Initialize** – The next step in the OTAP process is to reboot all chosen nodes to OTAP Image for preparation of OTAP'ing a new image. To do this, check the required nodes and click on **Initialize.** Alternatively, you can type the node IDs in **Select Nodes** box. Once the nodes are successfully initialized, the color of the selected nodes turns to Yellow as shown in Figure 8-6.

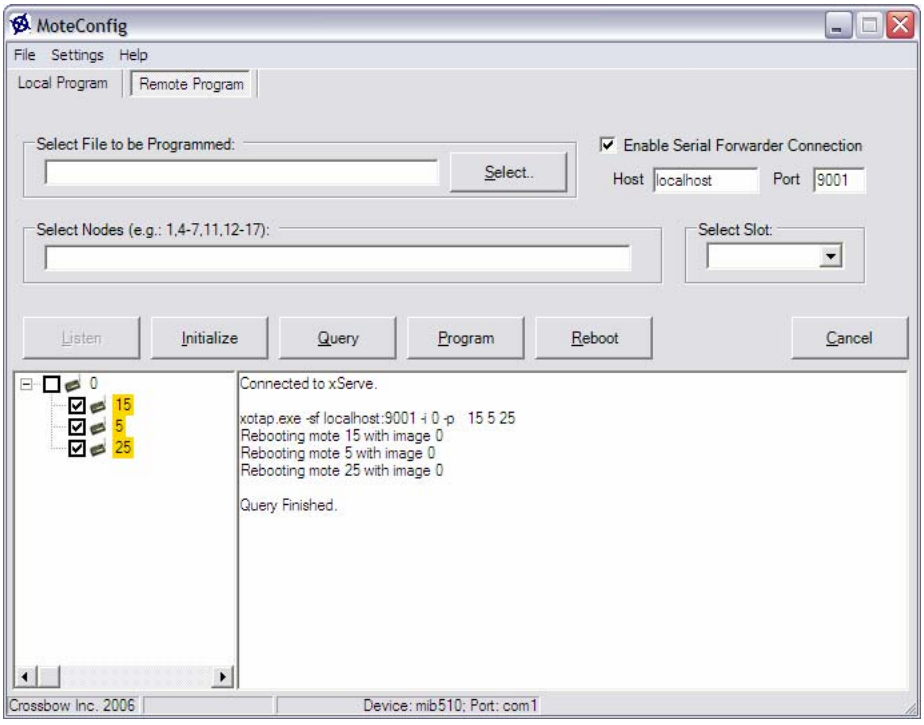


Figure 8-6. Initialization of Nodes in the Network for OTAP

4. **Query** – Users can optionally query a node when it is running the OTAP Image to get information about different slots as shown in Figure 8-7.

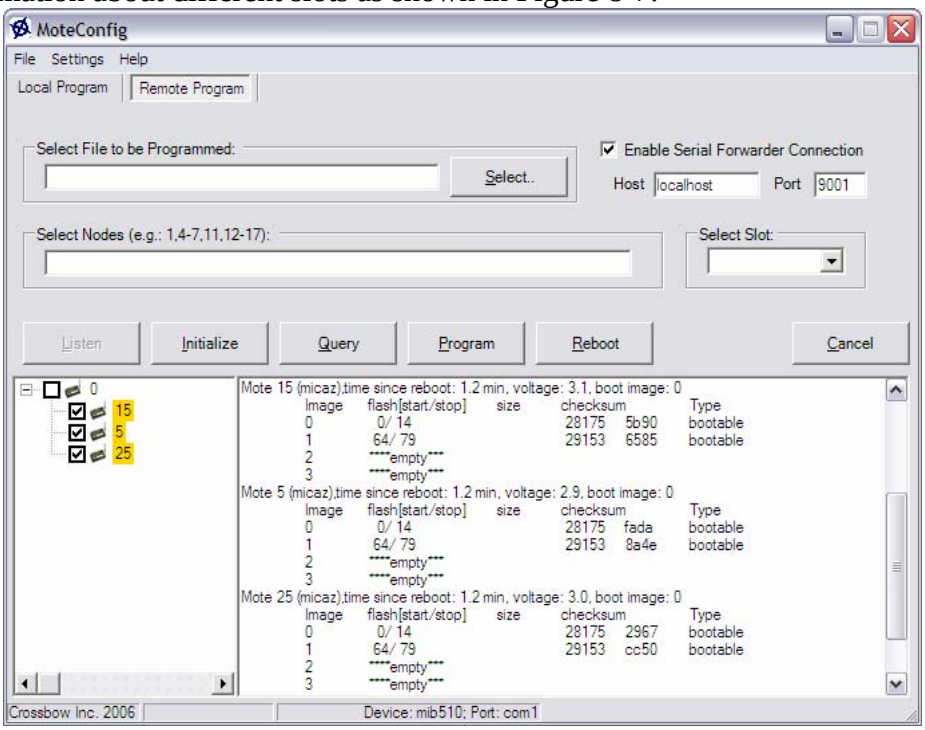


Figure 8-7. Querying the Nodes in the Network for OTAP Image

5. **Program** – Click on **Select**.. and browse to the binary image of the *XMesh* application (main.exe) that you want to program with. Check the nodes you want to program and specify the slot where you want to store the application. Alternatively, you can type the node IDs in **Select Nodes** box. Click on **Program**. As the OTAP progresses, you will see the report in terms of number of pages downloaded into the flash. During the OTAP process, the color of the selected nodes turns to Orange as shown in Figure 8-8.

⚠ **WARNING:** You cannot write over the Slot 0, because it is reserved for the OTAP Image.

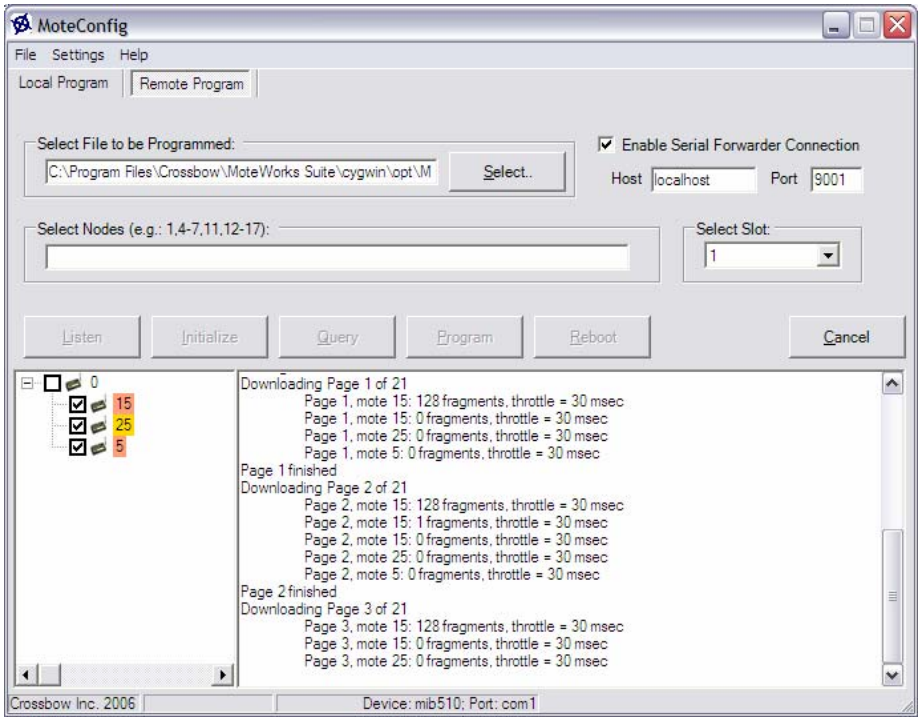

MoteConfig
File Settings Help
Local Program
Remote Program
Select File to be Programmed:
C:\Program Files\Crossbow\MoteWorks Suite\cygwin\opt\M
Select..
Enable Serial Forwarder Connection
Host localhost
Port 9001
Select Nodes (e.g.: 1,4-7,11,12-17):
Select Slot:
1
Listen
Initialize
Query
Program
Reboot
Cancel
0
15
25
5
Downloading Page 1 of 21
Page 1, mote 15: 128 fragments, throttle = 30 msec
Page 1, mote 15: 0 fragments, throttle = 30 msec
Page 1, mote 25: 0 fragments, throttle = 30 msec
Page 1, mote 5: 0 fragments, throttle = 30 msec
Page 1 finished
Downloading Page 2 of 21
Page 2, mote 15: 128 fragments, throttle = 30 msec
Page 2, mote 15: 1 fragments, throttle = 30 msec
Page 2, mote 15: 0 fragments, throttle = 30 msec
Page 2, mote 25: 0 fragments, throttle = 30 msec
Page 2, mote 5: 0 fragments, throttle = 30 msec
Page 2 finished
Downloading Page 3 of 21
Page 3, mote 15: 128 fragments, throttle = 30 msec
Page 3, mote 15: 0 fragments, throttle = 30 msec
Page 3, mote 25: 0 fragments, throttle = 30 msec
Crossbow Inc. 2006
Device: mib510; Port: com1

Figure 8-8. Programming the Nodes in the Network over the Air

When all the selected nodes are successfully programmed, you will see the output shown in Figure 8-9 and the color of the selected nodes turns back to Yellow.

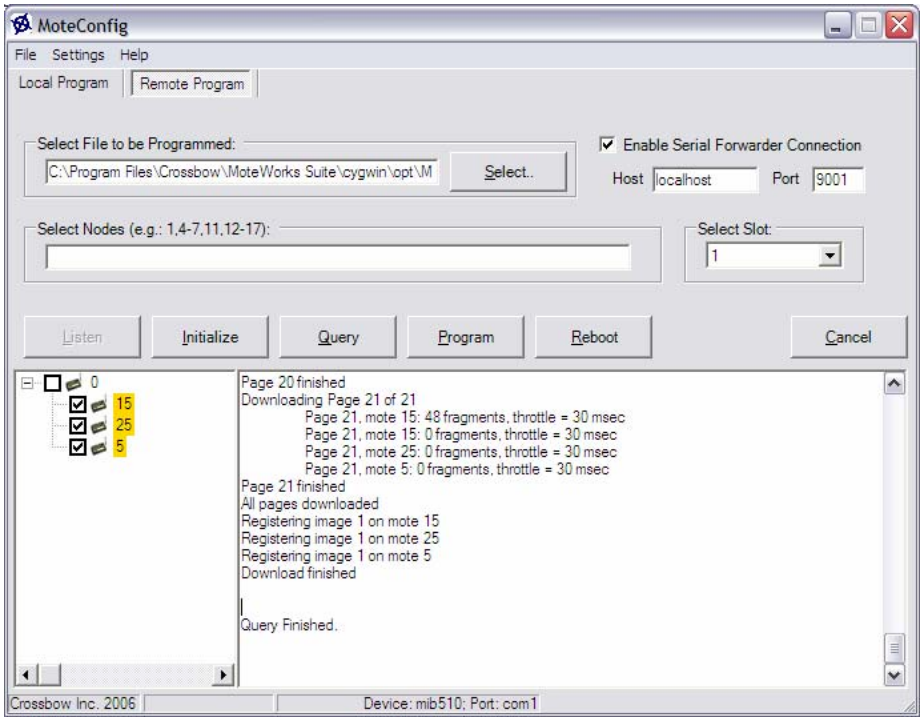


Figure 8-9. MoteConfig Screenshot after the Completion of programming over the Air

6. **Reboot** – The last step is to reboot into the newly loaded image for the selected nodes. To do this, check the nodes you want to reboot, specify the Slot and click on **Reboot**. Alternatively, you can type the node IDs in **Select Nodes** box. When all the selected nodes are successfully rebooted, you will see the output shown in Figure 8-10 and the color of the selected nodes turns back to Green.

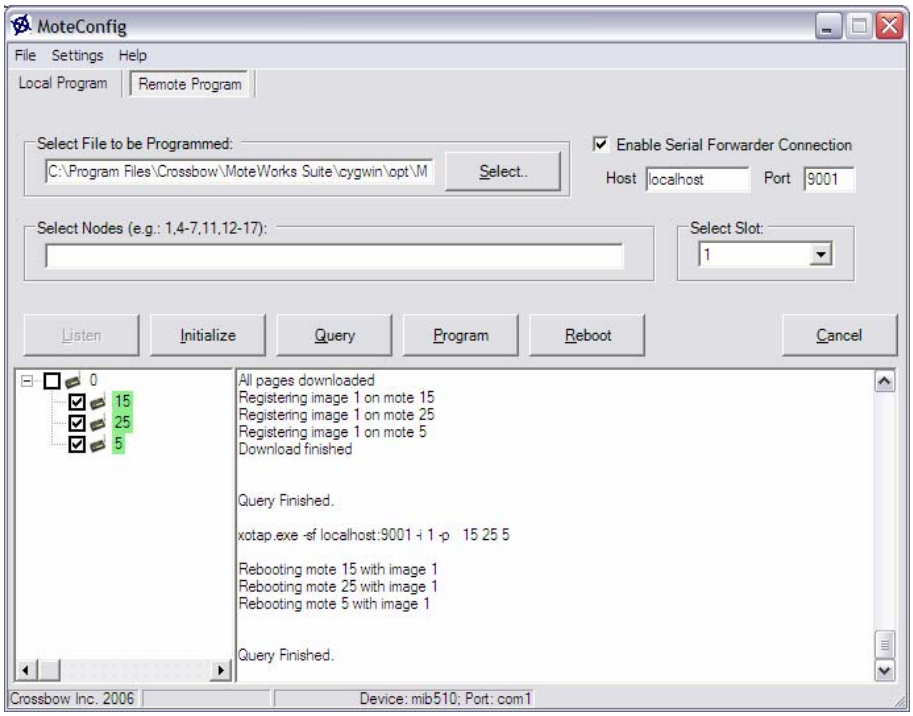


Figure 8-10. Rebooting the Nodes in the Network

The specified nodes should now start running the new application.

Crossbow Technology, Inc.
4145 N. First Street
San Jose, CA 95134
Phone: 408.965.3300
Fax: 408.324.4840